



# Введение в тестирование ПО

История тестирования,  
Тестирование и качество,  
Этапы тестирования

# Общая информация

## **Виды занятий и контроля по дисциплине:**

- Лекции – 1 пара в неделю
- Лабораторные работы – 1 пара в неделю
- Контрольные работы – 2 (на контрольных неделях)
- Зачет с оценкой

# Общая информация

## Рекомендуемая литература

- С. Макконнелл. Совершенный код. — СПб: «Питер», 2010. — 896 с.
- Г. Майерс. Надежность программного обеспечения. — М.: «Мир», 1980. — 360 с.
- [svyatoslav.biz/software\\_testing\\_book](http://svyatoslav.biz/software_testing_book) - Тестирование программного обеспечения. Базовый курс.
- [intuit.ru/studies/courses/48](http://intuit.ru/studies/courses/48) - курс лекций «Основы тестирования программного обеспечения»
- [istqb.org/downloads/glossary.html](http://istqb.org/downloads/glossary.html) (ISTQB - International Software Testing Qualifications Board)
- [sorlik.ru/swebok-ru/](http://sorlik.ru/swebok-ru/) (SWEBOK - Software Engineering Body of Knowledge)
- [software-testing.ru](http://software-testing.ru) – библиотека, статьи, ...

# История тестирования

# Что такое тестирование?

Для начала мы ...



... удостоверимся, **все ли в порядке**

# Тестирование необходимо?

- Тестирование необходимо, потому что **люди склонны ошибаться**. Одни ошибки незначительны, другие же опасны и дорого обходятся
- Поскольку ошибки допускают все люди, мы должны внимательно проверять результаты своей (и чужой) работы, всего, что мы делаем



# Зачем нужны тестировщики ПО?

- Должны найти и задокументировать ошибки до того как их найдут пользователи
- «Смотрят на продукт глазами пользователя» и проверяют основные сценарии использования продукта
- Обладают знаниями и навыками позволяющими проектировать и выполнять эффективные тесты
- Предоставляют заинтересованным сторонам информацию, достаточную для принятия обоснованного решения о готовности тестируемого продукта, передаче на следующий этап разработки или в качестве готовой системы пользователям

# С чего всё начиналось?

- только крупные фирмы и институты
- всего единицы продуктов в год
- сверхпрофессиональность кадров
- низкоуровневые языки разработки ПО

# Появление ПК – революция в области разработки ПО

- Проникновение компьютеризации во все сферы жизнедеятельности
- Увеличение количества фирм-разработчиков
- Постоянный рост создаваемых программ
- Пересмотр подхода к обеспечению качества и надежности программ

# Рост числа фирм - рост конкуренции

- Потребитель выбирает разработчика, обещающего оптимальное сочетание цены, времени разработки, качества продукта.
  - Разработчик ищет новые способы обхода конкурентов
  - Тестирование как конкурентное преимущество
- Всё больше и больше программ в наше время должны быть безупречными - Современные программы управляют оборудованием больниц, аэропортов, атомных реакторов, космических кораблей*

# Основные «эпохи тестирования»

- В **50–60-х годах** тестирование представляло собой скорее отладку программ (debugging) - *процесс поиска, анализа и устранения ошибок в ПО*
- Существовала концепция «исчерпывающего тестирования (exhaustive testing)» — проверки всех возможных путей выполнения кода со всеми возможными входными данными
- Однако очень скоро было выяснено, что исчерпывающее тестирование невозможно, т.к. количество возможных путей и входных данных очень велико, а также при таком подходе сложно найти проблемы в документации

# Задача о треугольнике

*По трём введённым целым числам определяет, может ли существовать треугольник с такими длинами сторон*

- Пусть стороны – тип `int` , т.е. 4 байта
- Тестовая система может проверить по 1 млн. проверок в секунду
- Сколько времени займет исчерпывающее тестирование?
- Как подготовить проверочные данные (верно ли сработала программа)?

# Основные «эпохи тестирования»

- В **70-х годах** фактически родились две фундаментальные идеи тестирования: тестирование - процесс доказательства работоспособности программы в некоторых заданных условиях (positive testing), а затем — строго наоборот: процесс доказательства неработоспособности программы в некоторых заданных условиях (negative testing)
- **!!! Неверное утверждение:** Негативные тест-кейсы должны заканчиваться возникновением сбоев и отказов в приложении
- **!!! Верное утверждение:** Негативные тест-кейсы пытаются вызвать сбои и отказы, но корректно работающее приложение выдерживает это испытание и продолжает работать верно

# Основные «эпохи тестирования»

- В **80-х годах** произошло ключевое изменение места тестирования в разработке ПО: вместо одной из финальных стадий создания проекта тестирование стало применяться на протяжении всего цикла разработки
- А также появились первые попытки автоматизировать тестирование
- В **90-х годах** произошёл переход от тестирования как такового к более всеобъемлющему процессу, который называется «*обеспечение качества (quality assurance)*», охватывает весь цикл разработки ПО и затрагивает процессы планирования, проектирования, создания и выполнения тест-кейсов, поддержку имеющихся тест-кейсов и тестовых окружений

# Основные «эпохи тестирования»

- В **2000-х годах** развитие тестирования продолжалось в контексте поиска всё новых путей, методологий, техник и подходов к обеспечению качества. Серьёзное влияние на понимание тестирования оказало появление *гибких методологий* разработки (подход TDD, test-driven development)
- Автоматизация тестирования уже воспринималась как обычная неотъемлемая часть большинства проектов, а также стали популярны идеи о том, что во главу процесса тестирования следует ставить не соответствие программы требованиям, а её способность предоставить конечному пользователю возможность эффективно решать свои задачи

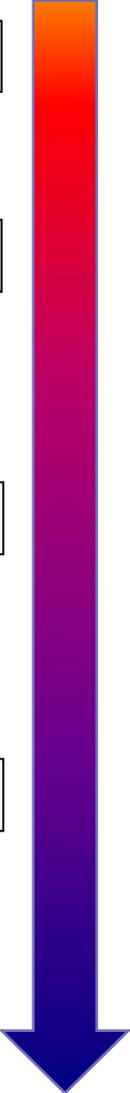
# Основные «эпохи тестирования»

На **современном этапе** развития тестирование имеет такие основные характеристики:

- гибкие методологии и гибкое тестирование
- глубокая интеграция с процессом разработки
- широкое использование автоматизации
- колоссальный набор технологий и инструментальных средств
- кросс-функциональность команды



# Что такое тестирование?



1979

- Это процесс исполнения программы с целью обнаружения ошибок (“Искусство тестирования программ”, Г. Майерс, 1979)

1987

- Процесс наблюдения за выполнением программы в специальных условиях и вынесения на этой основе оценки каких-либо ее аспектов (ANSI/IEEE standard 610.12-1990: Glossary of SE Terminology, 1987)

1999

- Техническое исследование программы для получения информации о ее качестве с точки зрения определенного круга заинтересованных лиц (С. Kaner, 1999)

2004

- Проверка соответствия между реальным поведением программы и ее ожидаемым поведением на конечном наборе тестов, выбранном определенным образом (IEEE Guide to Software Engineering Body of Knowledge, SWEBOK, 2004)

# Определение тестирования (классика)

- **Глен Маерс:**

Тестирование это процесс выполнения программ с намерением найти ошибки

- **Пол Йоргенсен:**

Тестирование сфокусировано на ошибках и сбоях. Тест – выполнение действий над ПО с целью найти ошибки или продемонстрировать работоспособность



*99 МАЛЕНЬКИХ БАГОВ В КОДЕ,  
99 МАЛЕНЬКИХ БАГОВ В КОДЕ,  
ОДИН НАШЛИ, ПОФИКСИЛИ.*

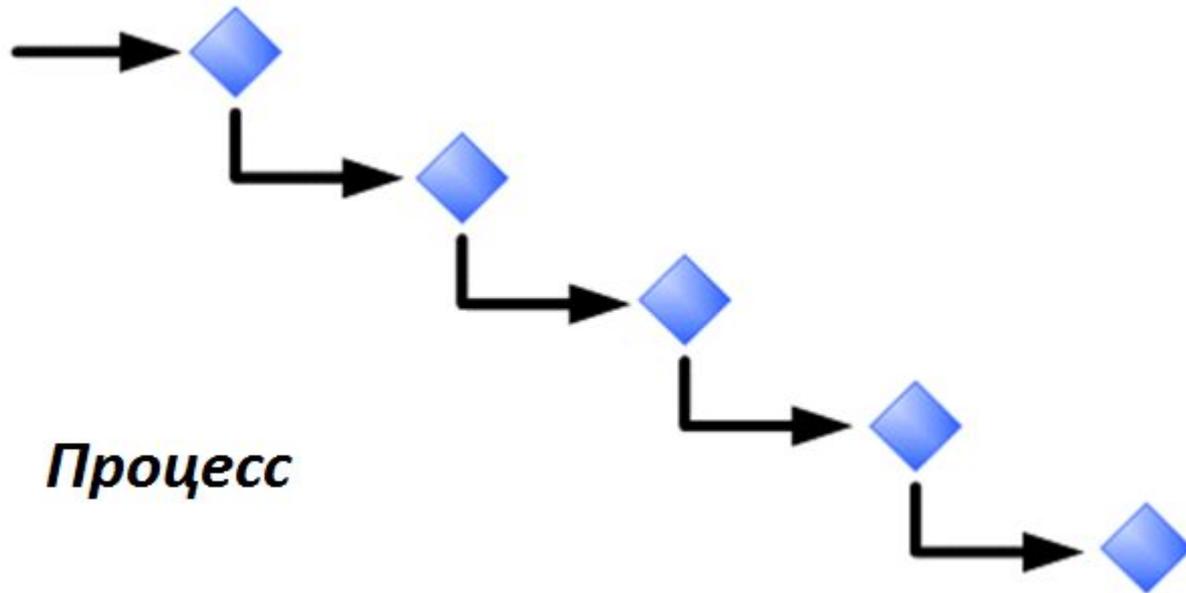
*127 МАЛЕНЬКИХ БАГОВ В КОДЕ*

# Определение тестирования

- процесс, содержащий в себе все активности жизненного цикла, как динамические, так и статические, касающиеся планирования, подготовки и оценки программного продукта и связанных с этим результатов работ с целью определить, что они соответствуют описанным требованиям, показать, что они подходят для достижения заявленных целей, а также для нахождения дефектов

# Определение тестирования «по частям» (1/5)

Тестирование – это **процесс**, а не единичное действие



# Определение тестирования «по частям» (2/5)

Процесс тестирования включен во **все**  
**активности жизненного цикла**



*Все активности  
жизненного цикла*

# Определение тестирования «по частям» (3/5)

Тестирование ПО может быть статическим и динамическим

- **Статическое** тестирование: Тестирование компонента или системы на уровне спецификации или реализации **без исполнения кода ПО**
- **Динамическое** тестирование: Тестирование, проводимое **во время выполнения** ПО, компонента или системы

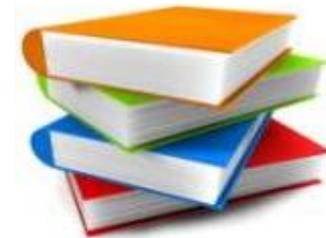
# Определение тестирования «по частям» (4/5)

- Планирование
- Подготовка
- Оценка



# Определение тестирования «по частям» (5/5)

Тестированию подлежит программный продукт и **связанные с ним рабочие продукты**



# Цели тестирования



- Предоставление информации для принятия решений
- Повышение уверенности в уровне качества
- Обнаружение дефектов
- Предотвращение дефектов

Тестирование помогает уменьшить общий уровень риска в системе после обнаружения и устранения дефектов и порождает уверенность в качестве ПО

# Объекты тестирования (1/2)

**Тестировать можно (и нужно!) всё:**

- работу программы
- качество ее кода и понятность комментариев
- быстродействие
- устойчивость под большой нагрузкой
- расход ресурсов (памяти, диска, потери этих ресурсов)

# Объекты тестирования (2/2)

- взаимодействие с другими программами
- стабильность работы
- возможность работы на других платформах
- удобство интерфейса
- документацию к программе (смысловые и грамматические ошибки, понятность и полноту)
- работу через сеть, работу аппаратного обеспечения и т.п.

# Тестирование и качество

# Тестирование и качество

- Что такое качество?



- «Качество – это ценность для индивидуума...» (Дж. Вайнберг, 1992)

- **Качество ПО** - это совокупность характеристик ПО, относящихся к его способности удовлетворять установленные и предполагаемые потребности

# Характеристики качества (ISO/IEC 9126)

- Функциональные возможности
- Функциональная пригодность
- Правильность (корректность)
- Способность к взаимодействию
- Надежность
- Эффективность
- Защищенность
- Портативность (мобильность)
- Практичность
- Удобство использования
- Удобство сопровождения



# Функциональность (1/10)

- **Функциональные возможности** — способность программного средства обеспечивать решение задач, удовлетворяющих сформулированным потребностям заказчиков и пользователей при применении комплекса программ в заданных условиях
- **Функциональная пригодность** — набор атрибутов, определяющих назначение, номенклатуру, основные, необходимые и достаточные функции программного средства, соответствующие техническому заданию и спецификациям требований заказчика или потенциального пользователя

# Правильность (2/10)

- **Правильность (корректность)** – способность программного средства обеспечивать правильные или приемлемые для пользователя результаты и внешние эффекты

# Взаимодействие (3/10)

- **Способность к взаимодействию** — свойство программных средств и их компонентов взаимодействовать с одним или большим числом компонентов внутренней и внешней среды

# Надежность (4/10)

- **Надёжность** – обеспечение комплексом программ достаточно низкой вероятности отказа в процессе функционирования программного средства в реальном времени
- Надежность – способность быть отказоустойчивым
- После всех проверочных работ тестировщика отказ оборудования является маловероятным событием

# Эффективность (5/10)

- **Эффективность** – свойства программного средства, обеспечивающие требуемый уровень производительности решения функциональных задач, с учётом количества используемых вычислительных ресурсов в установленных условиях
- Выдает оптимальное сочетание производительности приложения и затрачиваемых ресурсов системы

# Защищенность (6/10)

- **Защищённость** – способность компонентов программного средства защищать программы и информацию от любых негативных воздействий

# Портативность (мобильность) (7/10)

- **Мобильность** - подготовленность программного средства к переносу из одной аппаратно-операционной среды в другую
- По-другому - переносимость между операционными системами, оборудованием в целом

# Практичность (8/10)

- **Практичность (применимость)** – свойства программного средства, обуславливающие сложность его понимания, изучения и использования, а также привлекательность для квалифицированных пользователей при применении в указанных условиях

# Удобство использования (9/10)

- Легкость понимания, изучения, использования (логичность построения, интуитивность интерфейса)
- Привлекательность продукта (соответствие новым веяниям в графическом представлении)

# Удобство сопровождения (10/10)

- **Сопровождаемость** — приспособленность программного средства к модификации и изменению конфигурации и функций
- Добавление новой функциональности — обычное дело для приложений в промышленной эксплуатации
- Важно, чтобы приложение не имело внутренних барьеров против развития

# Тестирование и качество

- В IT-индустрии широко используется два понятия, напрямую связанные с тестированием ПО:
  - контроль качества  
(quality control, QC)
  - обеспечение качества  
(quality assurance, QA)



# Тестирование и качество

- Тестирование - это необходимый, но недостаточный этап обеспечения качества ПО



# Тестирование и качество

В **контроль качества** входят:

- *Тестирование*
- Рецензирование кода
- Статический анализ кода
- Внешняя оценка и аудит

В **обеспечение качества** входят:

- Усовершенствование процессов
- *Контроль качества*
- Управление изменениями

# Тестирование и качество

- *Как тестировщик может повлиять на качество?*
- Тестирование – возможный способ оценки качества ПО в терминах найденных дефектов, исполненных тестов и протестированных систем
- Когда во время тестирования находятся ошибки, качество ПО повышается, если эти дефекты исправлены

# Этапы тестирования

# Этапы тестирования

- Анализ требований
- Планирование испытаний
- Проектирование тестов
- Запуск тестов
- Редактирование тестов
- Системное тестирование
- Приемочные испытания
- Эксплуатация и сопровождение

# Анализ требований (1/8)

- Изучаем спецификации требований
- Изучаем функциональные требования к системе
- Отвечаем на вопросы:
  - что нам предстоит тестировать
  - как много будет работы
  - какие есть сложности
  - всё ли необходимое у нас есть и т.п.
- Получаем данные, по которым далее составляем план проведения испытаний

# Планирование испытаний (2/8)

- Определяем объемы испытаний и ресурсы
- Пишем расписание того, когда будем выполнять намеченные действия

# Проектирование тестов (3/8)

- Определяем:
  - Цель тестирования
  - Спецификацию входных данных
  - Архитектуру тестов (для упорядочения по группам)
- Пишем тесты

# Запуск тестов (4/8)

- Проверяем наши тесты в действии
- Анализируем тестовые случаи

# Редактирование тестов (5/8)

- Пересматриваем и корректируем тесты, т.к. в тестах тоже могут быть ошибки

# Системное тестирование (6/8)

- Проверяем всю систему
- Получаем сведения о качестве характеристик ПО

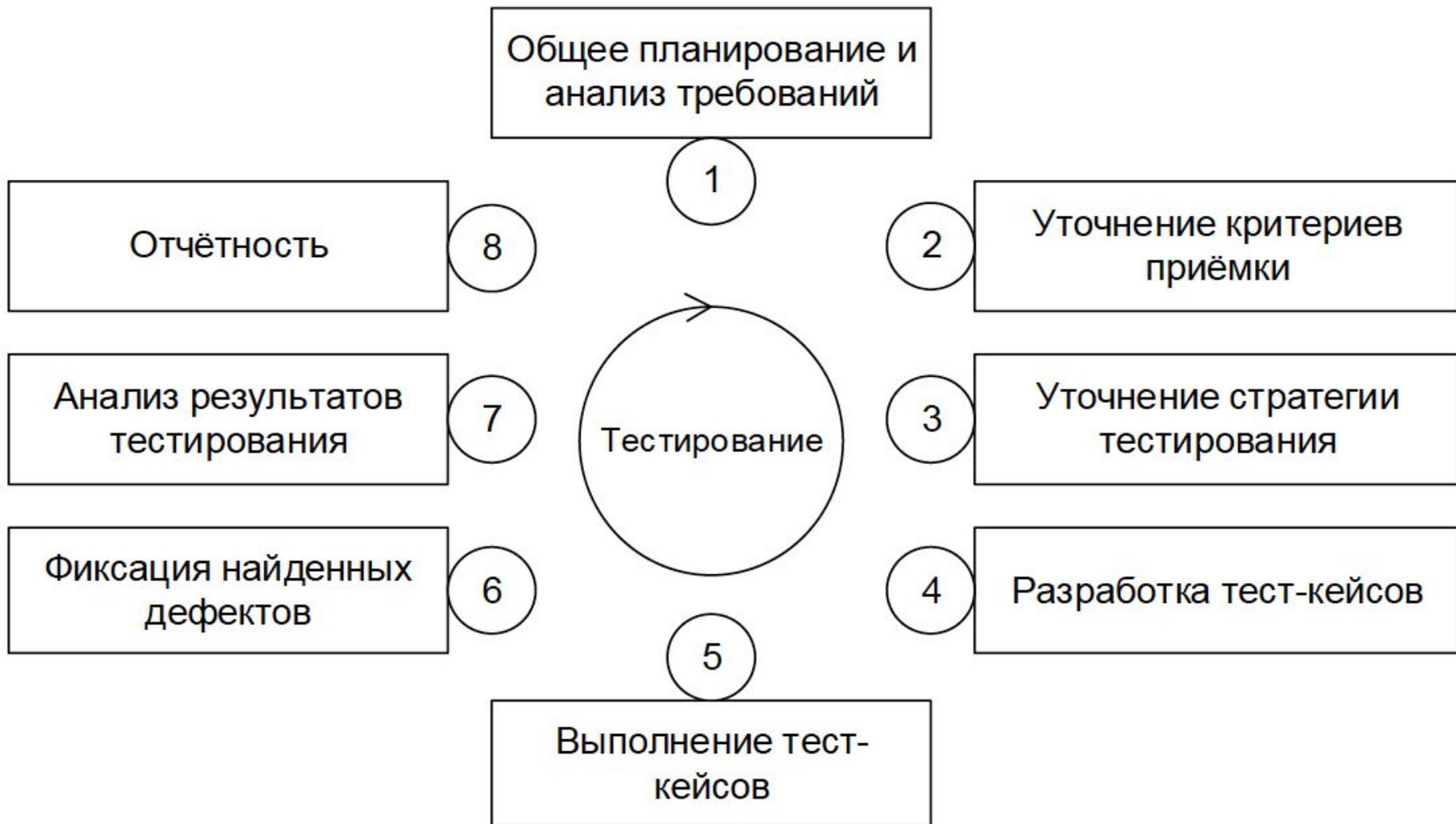
# Приемочные испытания (7/8)

- Альфа-тестирование
- Бета-тестирование

# Поддержка и сопровождение (8/8)

- Проверяем качество исправлений дефектов
- Проводим регрессионные тесты

# Жизненный цикл тестирования





До тестирования



После тестирования

Не бывает совершенных программ.  
Бывают неотестированные.