Виды, типы и области тестирования

Что должен делать тестировщик

- Сохранять здравый рассудок в творческой обстановке ИТ-проекта
- Хорошо знать продукт и то, как его используют клиенты
- Находить ошибки в продукте
- Помогать разработчикам исправлять ошибки
- Тестировать проектную документацию
- Предоставлять руководству всю необходимую для принятия решений информацию
- Оценивать качество продукта
- Выявлять способы улучшения продукта
- И много-много других не менее важных задач!

Тестирование, QC, QA

- Тестирование – исследование продукта и предоставление информации (дефекты, отчеты, метрики)
- Контроль качества (QC, Quality Control) – Тестирование + принятие решения о выпуске продукта.
- Обеспечение качества (QA, Quality Assurance) – процессный менеджмент, определяющий пути повышения качества продукта (не только в области тестирования).
- Валидация – верификация + тестирование требований.

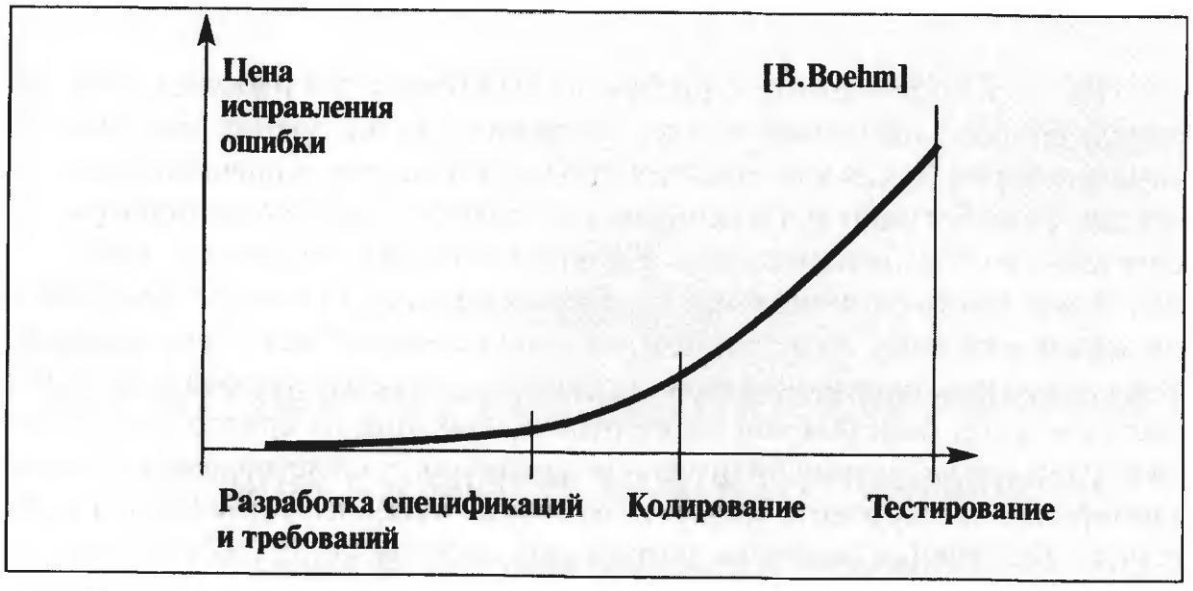
Тестирование, QC, QA

- **QA** – профилактика (терапевт)
- **QC** – оценка (диагност)
- **Testing** – лечение (помощник хирурга)

Стоимость исправления ошибок

Широко известна оценка распределения трудоемкости между фазами создания программного продукта: 40%-20%-40%

Миссия тестирования – снизить стоимость разработки путем раннего обнаружения дефектов



Самые худшие баги человечества

1962 г. Космический аппарат Mariner I стартовал по направлению к Венере. Корабль перешел на собственную систему пилотирования. Но эта система содержала обидный маленький баг. В результате аппарат полетел совсем не в ту сторону и его пришлось подорвать над Атлантическим океаном. Последующее расследование установило, что в процессе программирования системы навигации была совершена маленькая опечатка - при вводе одной из формул был пропущен один символ.

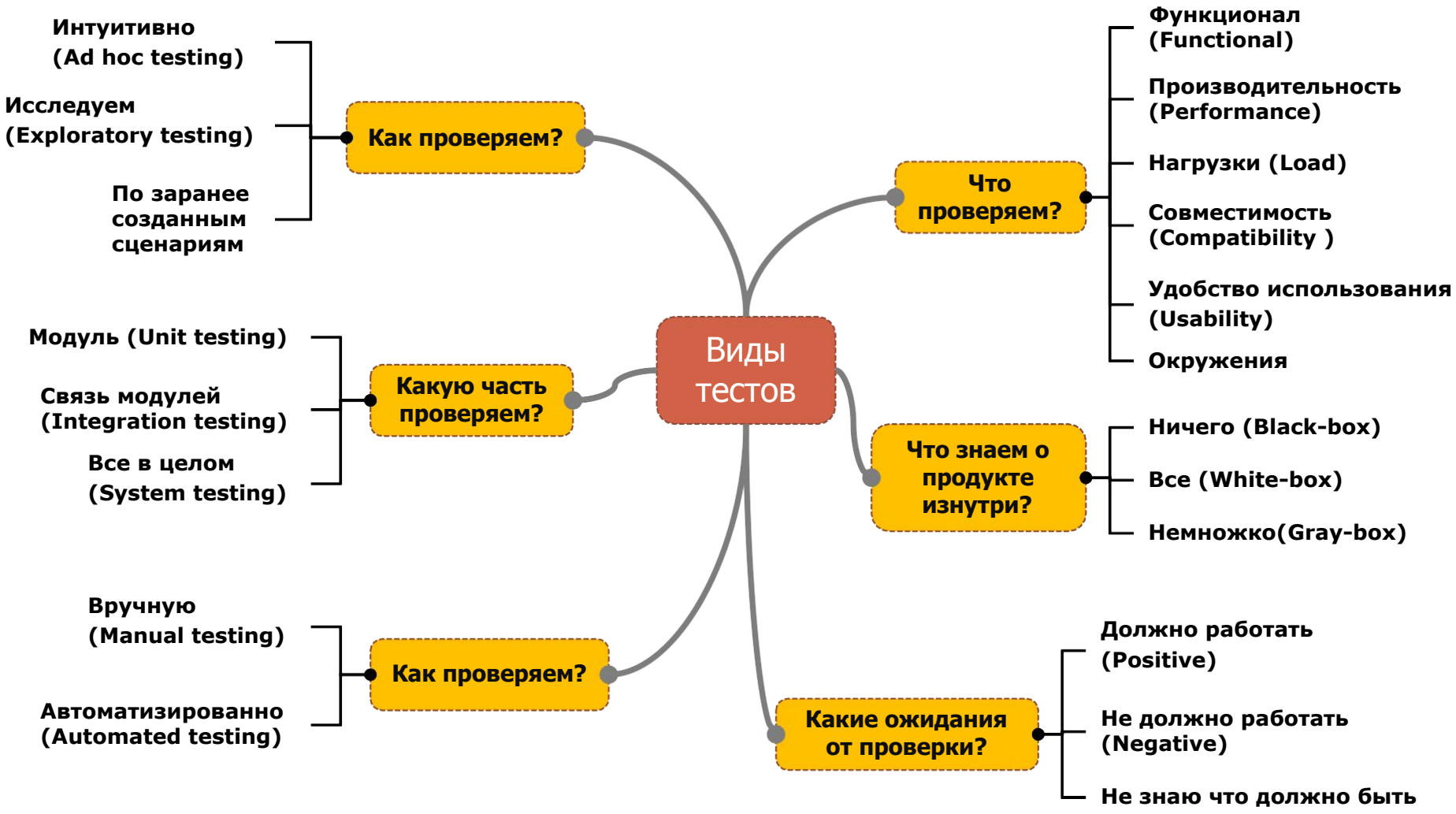
1982 г. Авария на Транссибирском трубопроводе. Оперативники ЦРУ внедрили баг (отчет в формате PDF) в канадское программное обеспечение, управлявшее газовыми трубопроводами. Советская разведка получила это ПО как объект промышленного шпионажа и внедрила на Транссибирском трубопроводе. Результатом стал самый большой неядерный взрыв в истории человечества.

1993 г. Процессор Intel Pentium неправильно производил деление с плавающей запятой, ошибаясь на 0,006%. Хотя эта проблема реально коснулась немногих пользователей, но стала настоящим кошмаром для имиджа Intel. Поначалу фирма согласилась менять процессор только для тех пользователей, которые могли доказать, что им в вычислениях нужна подобная точность, но затем согласилась поменять процессор всем желающим. Этот баг стоил Intel около $475 млн.

1996 г. Новая ракета-носитель Ariane 5, результат многолетней работы европейских ученых, гордость стран Евросоюза, взорвалась через 40 секунд после старта. Переполнение буфера, система навигации подала недопустимо большое значение параметра горизонтальной скорости. Инженеры сняли защиту от ошибок переполнения буфера в программном модуле, поскольку были уверены, что такого значения горизонтальной скорости не может быть в принципе. Только научное оборудование на борту ракеты стоило около $500 млн.

1985-87 гг. Несколько человек получили смертельную дозу облучения во время сеансов радиационной терапии с медицинским ускорителем Therac-25. «Улучшение" состояло в том, что вместо электромеханической защиты пациента в устройстве была реализована программная защита, якобы более надежная. Функция была некорректно реализована неопытным программистом, результатом чего стали как минимум пять смертей и огромное количество несмертельных случаев переоблучения.

Виды тестирования. Интеллект-карта



Позитивное и негативное тестирование

- **Позитивные**

  2+2=4

- **Негативные**

  10/0 = ошибка «На ноль делить нельзя»!

- **Исследовательские**

  Что будет, если перемножить максимально допустимые числа?

Функциональное и нефункциональное тестирование

- **Функционал**
  Добавленный товар попадает в корзину?
- **Производительность**
  Как быстро открывается страница?
- **Нагрузка**
  Могут ли 100 пользователей сделать заказ одновременно?
- **Окружения**
  Работает ли всё в Opera, IE, FF?
- **Совместимость**
  Работает ли магазин при использовании различных плагинов?
- **Удобство использования**
  Насколько понятно, как купить товар?

Модульное и интеграционное тестирование

- **Всю программу**

  ICQ функционирует как ожидается
- **Модуль**

  После регистрации в ICQ, в БД добавляется информация о пользователе (модуль регистрации)
- **Интеграцию модулей**

  Если добавить запись в условленном формате в БД, пользователь может войти в ICQ (интеграция регистрации и входа)

Альфа- и бета-тестирование

Альфа-тестирование – реальная работа с продуктом штатными сотрудниками команды разработки (программисты, тестировщики, аналитики, менеджер и т.д.) в роли конечных пользователей.

Бета-тестирование – использование почти готовой версии продукта группой сторонних пользователей с целью выявления максимального числа «нетипичных» дефектов для их исправления перед выходом продукта в релиз.

Бета-тестирования может использоваться как часть стратегии продвижения продукта на рынок:

- Бесплатная раздача бета-версий привлекает вниманием многих пользователей к окончательной, платной, версии.
- Сбор отзывов о продукте от широкого круга пользователей.

Регрессионное тестирование

Регрессионное тестирование (regression testing, от лат. regressio – движение назад) – собирательное название методов тестирования, направленных на обнаружение дефектов в уже протестированных частях продукта, которые не должны изменяться.

Такие ошибки появляются в результате влияния новых изменений на уже существующие части продукта.

Плохой дизайн и архитектура ПО – благодатная почва регрессионных дефектов.

"Фундаментальная проблема при сопровождении программ состоит в том, что исправление одной ошибки с большой вероятностью (20-50%) влечет появление новой". Фредерик Брукс - Мифический человеко-месяц или как создаются программные системы. (Премия Тьюринга 1999г.)

Приемочное тестирование

Приемочное тестирование (acceptance testing) – выполняется на основании набора типичных сценариев использования ПО, разработанных на основе требований к данному ПО.

Выполняется с целью демонстрации заказчику возможностей готового продукта. Заказчик принимает решение о принятии или отправлении на доработку продукта.

Smoke-тестирование

Smoke-тестирование – выполнение минимального набора тестов на явные ошибки.

Выполняется:

- Самим программистом после модификации кода. Если тест не пройдет, нет смысла отдавать продукт в глубокое тестирование
- В условиях критического недостатка времени на регрессионное тестирование при внесении небольших изменений

История термина:
Впервые использовался у печников Повторное рождение в электронике

Ad hoc тестирование

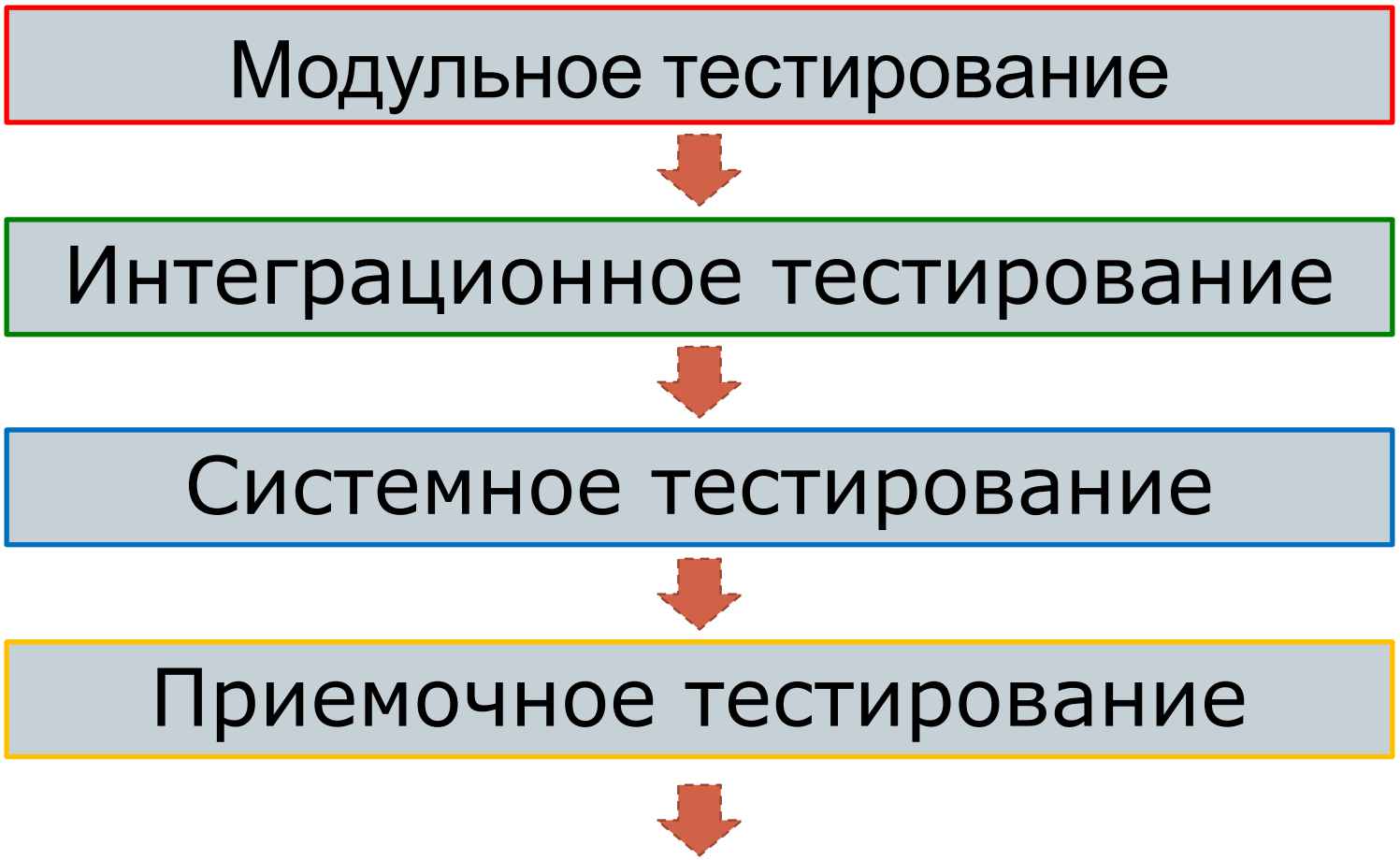
Ad hoc – (англ.) устроенный для данной цели.

Ad hoc тестирование – интуитивное тестирование, выполняемое без тест-кейсов, планирования и документации.

Выполняется:

- При приемке/сдаче продукта, если тесты не формализованы
- В довесок к документированному тестированию
- Код уже написан, нужно срочно протестировать

Связь уровней тестирования



Качество ПО

Тестирование - это возможный способ оценки качества программного обеспечения в терминах найденных дефектов.

Качество ПО (Quality) – характеристика ПО как степень соответствия ПО требованиям к нему.

По стандарту ISO 9001: **качество** есть степень соответствия присущих характеристик требованиям.

Говорят, что программный продукт обладает хорошим качеством, если :

- При работе пользователя с программным продуктом возникает небольшое число отказов .Этот факт свидетельствует о том, что на рабочее место просочилось лишь небольшое число дефектов .
- Программный продукт надежен, а это означает ,что его прогон редко завершается аварийным отказом или что он редко демонстрирует непредсказуемое поведение при работе в среде заказчика .
- Программный продукт удовлетворяет требованиям большинства пользователей .

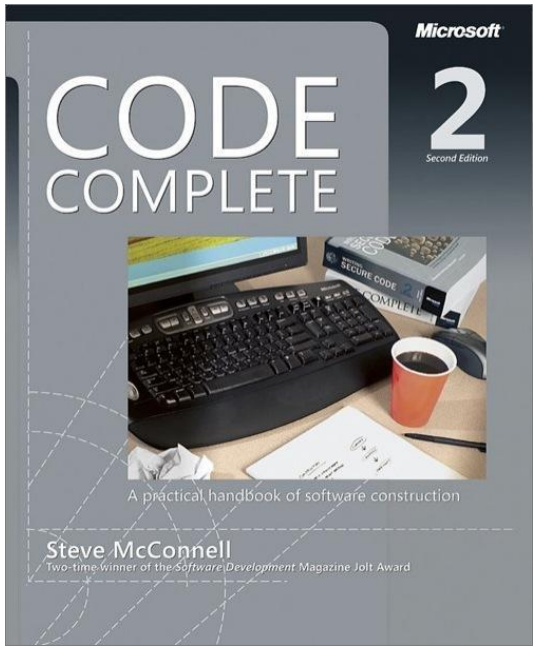
Критерии качества ПО

Тестирование является одним из наиболее устоявшихся способов обеспечения качества разработки программного обеспечения

- **Корректность**
  наличие/отсутствие дефектов в спецификации, проекте и реализации
- **Практичность**
  легкость изучения и использования
- **Эффективность**
  степень использования системных ресурсов
- **Надежность**
  способность системы выполнять необходимые функции; интервал между отказами
- **Целостность**
  способность предотвращать неавторизованный или некорректный доступ
- **Адаптируемость**
  возможность использования в других областях и средах
- **Правильность**
  степень безошибочности данных, выдаваемых системой
- **Живучесть**
  способность продолжать работу при недопустимых данных или в напряженных условиях

Критерии качества ПО

	корректность	практичность	эффективность	надежность	целостность	адаптируемость	правильность	живучесть
корректность	✓		✓	✓			✓	✓
практичность		✓				✓	✓	
эффективность	✓		✓	✓	✓	✓	✓	
надежность	✓			✓	✓		✓	✓
целостность			✓	✓	✓			
адаптируемость					✓	✓		✓
правильность	✓		✓	✓		✓	✓	✓
живучесть	✓	✓	✓	✓	✓	✓	✓	✓



Семь главных принципов тестирования

- **Тестирование демонстрирует наличие дефектов**
- **Исчерпывающее тестирование недостижимо**
- **Раннее тестирование**
- **Скопление дефектов**
- **Парадокс пестицида**
- **Тестирование зависит от контекста**
- **Заблуждение об отсутствии ошибок**