



Введение в тестирование

Новикова Анастасия
Ст.специалист

ВАШ СОЮЗНИК В РАЗВИТИИ

Тестирование программ можно использовать для того, чтобы показать наличие ошибок, и никогда — для того чтобы показать их отсутствие!

Эдсгер Дейкстра

Эволюция представлений о тестировании

-
- 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*]
 - 1999 • Техническое исследование программы для получения информации о ее качестве с точки зрения определенного круга заинтересованных лиц. [*C. Kaner, 1999*]
 - 1990 • Это не действие. Это интеллектуальная дисциплина, имеющая целью получение надежного программного обеспечения без излишних усилий на его проверку. [*B. Beizer. Software Testing Techniques, Second Edition. NY:van Nostrand Reinhold, 1990*]
 - 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. NY:IEEE, 1987*]
 - Процесс выполнения программы с намерением найти ошибки. [*Г. Майерс. Надежность программного обеспечения. М:Мир, 1980*]



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

 **Тестирование** – процесс проверки соответствия заявленных к продукту требований и реально реализованной функциональности, осуществляемый путем наблюдения за его работой в искусственно созданных ситуациях и на ограниченном наборе тестов, выбранных определенным образом



-  **Функциональные требования** регламентируют функционирование или поведение системы (behavioral requirements). Функциональные требования отвечают на вопрос "что должна делать система" в тех или иных ситуациях. Функциональные требования определяют основной "фронт работ" Разработчика, и устанавливают цели, задачи и сервисы, предоставляемые системой Заказчику.
-  **Нефункциональные требования**, соответственно, регламентируют внутренние и внешние условия или атрибуты функционирования системы

Что общее и в чем различие?




-  **QA** — это забота о качестве в виде предупреждения появления дефектов
-  **Тестирование** — это забота о качестве в виде обнаружения дефектов до того, как их найдут пользователи.
-  **Общее** в QA и тестировании заключается в том, что они призваны улучшить ПО
-  **Различие** между ними — в том, что QA призвано улучшить ПО через улучшение процесса разработки ПО, а тестирование — через обнаружение дефектов.

Тестировать можно все:







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



Что такое дефект?




-  Ожидаемый результат
-  Фактический результат
-  Несоответствие!



-  По степени доступа к системе
-  По объекту тестирования
-  По времени проведения тестирования
-  По критерию позитивности сценариев
-  По степени изолированности компонентов
-  По степени автоматизированности









По степени доступа к системе

-  Черный ящик (black box testing);
-  Серый ящик (grey box testing);
-  Белый ящик (white box testing).



 **Функциональное тестирование** (functional testing);

 **Нефункциональное тестирование:**

-  тестирование интерфейса пользователя (UI testing);
-  тестирование локализации (localization testing);
-  нагрузочное тестирование (performance testing);
-  тестирование безопасности (security testing);
-  тестирование эргономичности (usability testing);
-  тестирование совместимости (compatibility testing)

По времени проведения тестирования



Альфа-тестирование

✕ Выполняется группой тестирования внутри команды/организации разработки



Бета-тестирование

✕ Выполняется группой тестирования в среде дружелюбно настроенных клиентов



Регрессионное тестирование

✕ Выборочное тестирование, позволяющее убедиться, что изменения не вызвали нежелательных побочных эффектов, или что измененная система по-прежнему соответствует требованиям.



Приемочное тестирование

✕ Выполняется заказчиком с целью определить, будет ли система принята в эксплуатацию
Выполняется группой тестирования с целью определить, будет ли система принята в тестирование (smoke testing)

✘ **позитивное тестирование** (positive testing);

✘ **негативное тестирование** (negative testing).





модульное тестирование (component testing)

- ✘ Тестирование целостности кода на уровне логических модулей
- ✘ Выполняется разработчиками
- ✘ Контролируется группой тестирования с помощью инструментов анализа покрытия кода unit-тестами (unit test coverage tools)



интеграционное тестирование (integration testing);




- ✘ Тестирование промежуточных результатов интеграции системы
- ✘ Выполняется разработчиками и тестировщикам
- ✘ Возможны подходы «сверху вниз» и «снизу вверх»



системное тестирование (system testing).

- ✘ Проверка полностью построенной системы на соответствие сформулированным требованиям
- ✘ Подуровни:
 - ✘ Альфа-тестирование
 - ✘ Бета-тестирование
 - ✘ Приемочное тестирование

По степени автоматизации

-  **ручное** тестирование (manual testing);
-  **автоматизированное** тестирование (automated testing);
-  **смешанное/полуавтоматизированное** тестирование (semiautomated testing).



Главные должностные обязанности тестировщика

- Контроль качества разрабатываемых продуктов.
- Выявление и анализ ошибок и проблем, возникающих у пользователей при работе с программными продуктами.
- Разработка и прохождения сценариев тестирования.
- Документирование найденных дефектов.
- Разработка автотестов и их регулярный прогон.



До тестирования

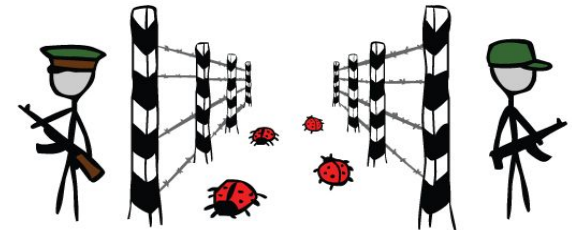


После тестирования

Не бывает совершенных программ.
Бывают неотестированные.

- Возможность влиять на качество.
- Участие на каждом этапе разработки ПО.
- Возможность принимать итоговые решения по требованиям и срокам, как заключительное звено в процессе разработки.
- Возможность видеть картину системно, а не смотреть только на свою часть.
- Работа в команде.
- Получаешь эстетическое удовольствие от борьбы за качество.
- Работа тестировщика всегда востребована.
- Это просто интересно.

Баги водятся на границах



1. С. Канер, Д. Фолк, Е. Нгуен. Тестирование программного обеспечения. — К.: Диасофт, 2000. — 544 с.
2. Р. Калбертсон, К. Браун, Г. Кобб. Быстрое тестирование. — М: Вильямс, 2002.
3. С. Макконнелл. Совершенный код. — СПб: «Питер», 2005. — 896 с.
4. Г. Майерс. Искусство тестирования программ. — М.: «Финансы и статистика», 1982. — 176 с.
5. Л. Тамре. Введение в тестирование программного обеспечения — М.: «Вильямс», 2003. — 368 с.
6. Г. Майерс. Надежность программного обеспечения. — М.: «Мир», 1980. — 360 с.
7. Б. Бейзер. Тестирование черного ящика. — СПб: «Питер», 2005. — 318 с.
8. Э. Брауде. Технология разработки программного обеспечения. — СПб: «Питер», 2004. — 655 с.
9. С. Орлов. Технологии разработки программного обеспечения. — СПб: «Питер», 2003. — 480 с.

10. И. Винниченко. Автоматизация процессов тестирования. — СПб: «Питер», 2005. — 203 с.
11. К. Бек. Экстремальное программирование. — СПб: «Питер», 2002.
12. К. Ауэр, Р. Миллер. Экстремальное программирование. — СПб: «Питер», 2003. — 368 с.
13. Д. Бентли. Жемчужины программирования. — СПб: «Питер», 2002. — 272 с.
14. С. Бобровский. Технологии Пентагона на службе российских программистов. — СПб: «Питер», 2003. — 222 с.
15. А. Якобсон, Г. Буч, Д. Рамбо. Унифицированный процесс разработки программного обеспечения. — СПб: «Питер», 2002. — 496 с.
16. Р. Мартин. Чистый код: создание, анализ и рефакторинг. — СПб: «Питер», 2010. — 464 с.

1. sorlik.ru/swebok-ru/
(SWEBOK - Software Engineering Body of Knowledge)
2. software-testing.ru – библиотека, статьи, ...
3. wiki.agiledev.ru/doku.php – гибкая разработка и тестирование
4. ru.wikipedia.org – Тестирование ПО, ISO 9126
5. www.intuit.ru/catalog/se/testing - курсы лекций
6. www.cmcons.com/map - карта сайта. Смотреть:
 1. Термины тестирования ПО; Термины, относящиеся к качеству
 2. Метрики кода; Тест Джоэла, ...
7. <http://www.osp.ru/os/2008/07/5478839/> (Б.Майер, 7 принципов тестирования ПО)

**БЛАГОДАРИМ ЗА ВНИМАНИЕ.
Задавайте, пожалуйста, вопросы.**

www.neoflex.ru

Телефон: +7 (495) 984-25-13, 984-27-90

Neoflex
ИТ-РЕШЕНИЯ ДЛЯ БИЗНЕСА 