Manual QA course

Lecture 8. Виды тестирования. Часть 2

Дорофеев Максим

hillel
компьютерная
школа

Функциональное тестирование.

Функциональное тестирование рассматривает заранее указанное поведение и основывается на анализе спецификаций функциональности компонента или системы в целом.

Функциональное тестирование.

Функциональные тесты основываются на функциях, выполняемых системой, и могут проводиться на всех уровнях тестирования (**компонентном, интеграционном, системном, приемочном**). Как правило, эти функции описываются в требованиях, функциональных спецификациях или в виде случаев использования системы (**use cases**).

Функциональное тестирование.

Тестирование функциональности может проводиться в двух аспектах:

- Требования;
- Бизнес-процессы.

Функциональное тестирование.

Тестирование в перспективе «**требования**» использует спецификацию функциональных требований к системе как основу для дизайна тестовых случаев **(Test Cases)**. В этом случае необходимо сделать список того, что будет тестироваться, а что нет, приоритезировать требования на основе рисков (если это не сделано в документе с требованиями), а на основе этого приоритезировать тестовые сценарии **(test cases)**. Это позволит сфокусироваться и не упустить при тестировании наиболее важный функционал.

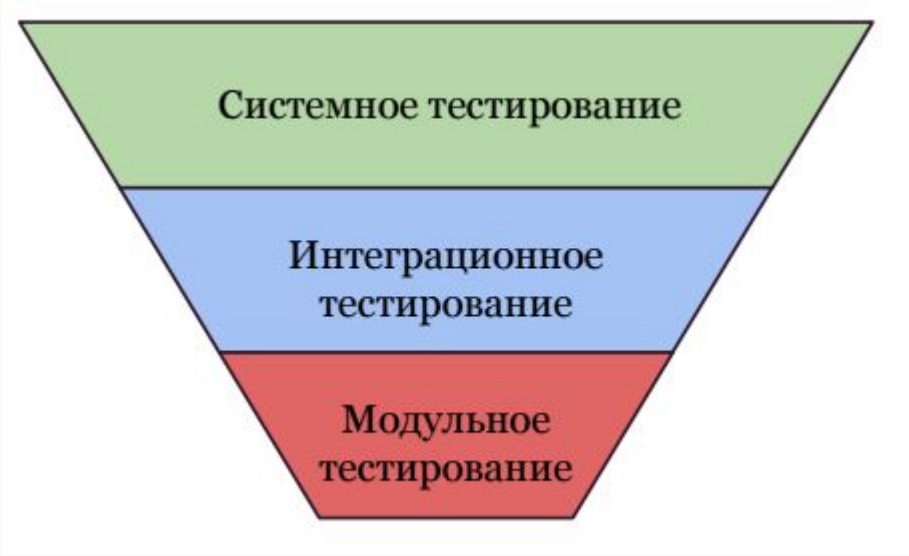
Функциональное тестирование.

Тестирование в перспективе «**бизнес-процессы**» использует знание этих самых бизнес-процессов, которые описывают сценарии ежедневного использования системы. В этой перспективе тестовые сценарии (**test scripts**), как правило, основываются на случаях использования системы (**use cases**).

Функциональное тестирование.

Максимальное приближение к реальным сценариям использования ПО.

Функциональное тестирование.



Может применяться

на всех уровнях

тестирования.

Функциональное тестирование.

Чаще всего является **формальным** тестированием,
т.е. основано на:

- Документации
- Описании ***user scenarios***

Функциональное тестирование.

Это всегда тестирование “чёрного ящика” (!)

Функциональное тестирование. Цели.

Test to pass:

- ПО минимально работоспособно;
- Простые сценарии;
- Не превышаем ограничения;
- Не ищем баги.

Функциональное тестирование. Цели.

Test to fail:

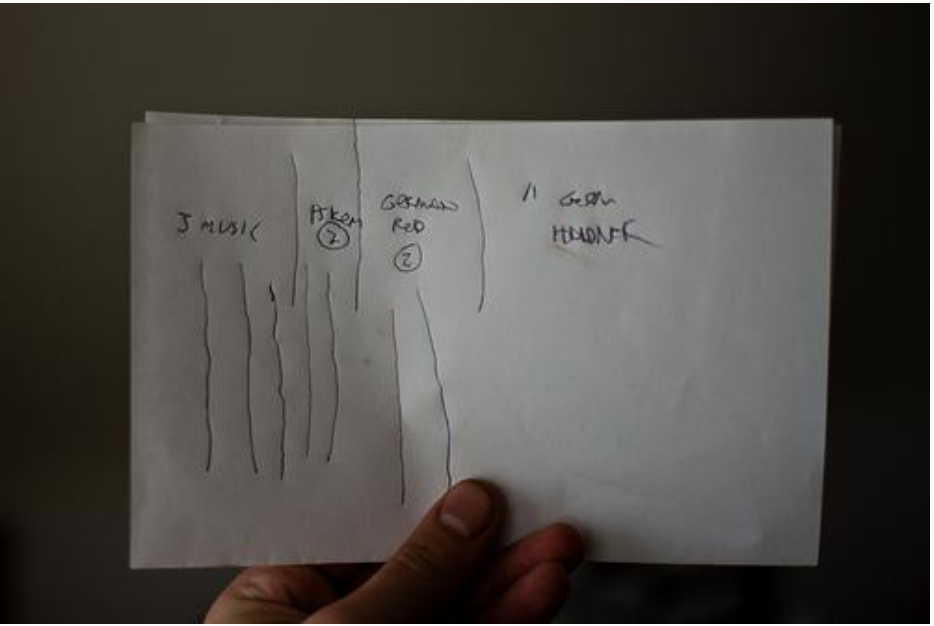
- Сценарии, которые могут сломать ПО;
- Известные и неизвестные слабые места;
- Акцент на поиск ошибок.

Функциональное тестирование. Достижения

Функциональное тестирование. Примеры.

- “Распечатать счет - фактуру”;
- “Показать фотографию на странице”;
- “Загрузить годовой отчет”;
- “Оплатить выбранные товары”.

Функциональное тестирование. Недостатки формального подхода.



- Документации нет;
- Документации мало;
- Документация плохого качества.

Функциональное тестирование. Недостатки формального подхода.

- Документации очень много.

Функциональное тестирование. Ручное против автоматизированного.

VS

Функциональное тестирование. Плюсы ручного тестирования.

- Легкая доступность для начинающих;
- Можно проверить очень сложные сценарии.

Функциональное тестирование. Минусы ручного тестирования.

- Устал;
- Забыл;
- Забил;
- Не подумал;
- Скорость и объем.

Функциональное тестирование. Плюсы автотестирования.

- Скорость;
- Исключаем человека;
- Автоматическая отчетность.

Функциональное тестирование. Минусы автотестирования.

- Высокий порог вхождения;
- Стоимость поддержки;
- Дополнительная инфраструктура;
- Часть сценариев не поддается автоматизации.

Функциональное тестирование.

Функциональное тестирование.

Плюсы функционального тестирования:

- Имитирует фактическое использование системы;

Минусы функционального тестирования:

- Возможность упущения логических ошибок в ПО;
- Вероятность избыточного тестирования.

Тестирование безопасности. Основная модель.

- Конфиденциальность;
- Целостность;
- Доступность.

Тестирование безопасности. Необязательные модели.

- Неотказуемость;
- Подотчетность;
- Достоверность;
- Аутентичность.

Тестирование безопасности. Зачем?

- Информация и контроль доступа;
- Стабильность системы;
- Целостность системы;
- Экономическая эффективность.

Тестирование безопасности. Где применяется?

- Приложения с важной коммерческой или персональной информацией;
- Платежные системы;
- Приложения требующие целостности информации;
- Социальные приложения;
- Приложения с коммерческим лицензированием

Тестирование безопасности. Особенности.

- Важность “негативного” тестирования;
- Думать как хакер;
- Качество тестирования безопасности сложно измерить;
- Важность нефункциональных требований;
- Тестирование на основе рисков;

Тестирование безопасности. Стандарты.

- OSSTMM (http://www.isecom.org);
- ISACA (http://www.isaca.org);
- ISSAF (http://www.oissg.org/issaf);
- OWASP Guide (https://www.owasp.org/);
- NIST Guideline (http://csrc.nist.gov/about/);

Тестирование безопасности. Типичные уязвимости.

- Неверная валидация входных данных;
- Внедрение параметров (XSS, CSRF);
- Переполнение буфера;
- Инъекции;
- Неверное завершение сессий;

Тестирование безопасности. Методы тестирования.

- **Построение модели угроз и рисков**;

Вероятность (возможность) реализации угрозы (Y_j)	Степень возможного ущерба (X_j)		
	Низкая	Средняя	Высокая
Низкая	неактуальная	неактуальная	актуальная
Средняя	неактуальная	актуальная	актуальная
Высокая	актуальная	актуальная	актуальная

Тестирование безопасности. Методы тестирования.

Поиск уязвимостей в исходном коде:

- Ревью кода разработчиками;
- Анализ кода при помощи утилит статического и динамического анализа;

Тестирование безопасности. Методы тестирования.

Тестирование на проникновение:

- Веб - сканеры;
- Анализ сетей;
- Ручное тестирование на проникновение.

Тестирование безопасности. Методы тестирования.

Нефункциональное тестирование:

- Нагрузочное тестирование;
- Стресс - тестирование;
- Объемное тестирование;
- Тестирование масштабируемости;

Тестирование безопасности. Сложности

- Приложение может вести себя по разному на различных платформах;
- Много конфигураций;
- Различное железо;
- Разные драйверы.

Тестирование безопасности. Этапы тестирования

- Сбор информации;
- Анализ угроз, уязвимостей, построение матриц угроз и рисков;
- Определение критериев защищенности, Простые тесты, анализ исходного кода;
- Внешняя экспертиза, нефункциональное тестирование, тестирование, основанное на рисках;
- Нагрузочные тесты, тестирование на проникновение.

Тестирование безопасности. Выводы

- Тестирование безопасности - необходимый этап, для компаний, которым важен “безопасный” продукт;
- Невозможно измерить качество тестирования;
- Необходимо изучать новые технологии;
- Необходимо регулярно проводить экспертизу;

Вопросы и ответы

Полезные ссылки

Functional testing

http://softwaretestingfundamentals.com/functional-testing/

OWASP

Ron Patton - Software testing