Topic 3. Fundamentals of Testing

Содержание:

1. Баг репорт.
2. Тест-план.
3. Тест-кейсы.
4. Виды тестирования.

Освежим:

1. Почему дефект может нанести ущерб человеку, оборудованию или компании?
2. В чем разница между причиной дефекта и его эффектом?
3. Примеры необходимости тестирования.
4. Разница между тестированием и QA.
5. Как тестирование способствует повышению качества.
6. Сравнить термины ошибка, дефект, отказ.
7. Вспомнить общие цели тестирования.
8. Примеры, отражающие цели тестирования на различных стадиях жизненного цикла программного обеспечения.
9. Отличия тестирование от отладки.
10. Вспомнить психологические факторы, которые влияют на успех тестирования.
11. Сравнить мышление тестировщика и разработчика.

Почему тестирование необходимо?


ФШ-ФШ-ФШ
Узри - я *джинн из лампы*!
Я могу исполнить
три твоих желания!
Каковое твое
первое желание?
Больше
желаний!
Джинн может исполнить
только три желания.
Таков закон.
Больше
джиннов!
Блин!
Я НАШЁЛ БАГ

Что такое тестирование?

Тестирование — это процесс, а не отдельное действие. Данный процесс начинается с планирования, затем проектирование тестов, подготовки к исполнению и оценки состояния до закрытия теста.

1. Testing shows presence of defects
Тестирование демонстрирует наличие дефектов.
2. Exhaustive testing is impossible
Исчерпывающее тестирование невозможно.
3. Early testing
Раннее тестирование.
4. Defects clustering
Кластеризация дефектов.
5. Pesticide Paradox
Парадокс пестицида.
6. Testing is context dependent
Тестирование зависит от контекста.
7. Absence of errors fallacy
Заблуждение об отсутствии ошибок!

1. Planning and Control
Планирование и контроль
2. Analysis and Design
Анализ и дизайн
3. Implementation and Execution
Реализация и выполнение
4. Evaluating exit criteria and Reporting
Оценка критериев выхода и составление отчетности
5. Test Closure activities
Действия по завершению тестирования

У БАГОВ ТОЖЕ ЕСТЬ ЧУВСТВА

НАЙДЯ БАГ:
СООБЩИТЕ О НЕМ

БАГИ НЕ ЛЮБЯТ БЫТЬ ЗАБЫТЫМИ

НАЙДЯ БАГ:
ИЗУЧИТЕ ЕГО

БАГИ ЛЮБЯТ БЫТЬ ПОНЯТЫМИ

НАЙДЯ БАГ:
СДЕЛАЙТЕ СНИМОК

БАГИ ЛЮБЯТ ХРАНИТЬ ПАМЯТЬ О ВСТРЕЧЕ

НАЙДЯ БАГ:
УЗНАЙТЕ ЕГО ДРУЗЕЙ

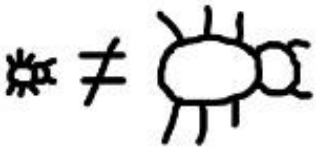
БАГИ ОБЩИТЕЛЬНЫ

НАЙДЯ БАГ:
СООБЩИТЕ О НЕМ БЫСТРО

ИНАЧЕ БАГИ ОБЖИВУТСЯ И ПОСТРОЯТ СЕБЕ ДОМА

НАЙДЯ БАГ:
БУДЬТЕ ЧЕСТНЫ

БАГИ НЕ ЛЮБЯТ СПЛЕТЕН



НАЙДЯ БАГ:
ЗАПОМНИТЕ КАК ВЫ ВСТРЕТИЛИ ЕГО

БАГИ РОМАНТИЧНЫ

НАЙДЯ БАГ:
НЕ ИГНОРИРУЙТЕ ЕГО

БАГИ МОГУТ УКУСИТЬ, ЕСЛИ ИХ НЕ ЦЕНИТЬ

AG

Атрибуты дефекта

BUG ID (НОМЕР БАГА)
SUMMARY (КРАТКОЕ ОПИСАНИЕ)
DESCRIPTION AND STEPS TO REPRODUCE (ОПИСАНИЕ)
ATTACHMENT (ПРИЛОЖЕНИЕ)
SUBMITTED BY (АВТОР БАГА)
DATE SUBMITTED (ДАТА И ВРЕМЯ ПОЯВЛЕНИЯ БАГА)
ASSIGNED TO (ДЕРЖАТЕЛЬ БАГА)
ASSIGNED BY (ИМЯ ПЕРЕДАВШЕГО БАГ)
COMPONENT (КОМПОНЕНТ)
COMMENTS (КОММЕНТАРИИ)
SEVERITY (СЕРЬЕЗНОСТЬ БАГА)
PRIORITY (ПРИОРИТЕТ БАГА)
NOTIFY LIST (СПИСОК ДЛЯ ОПОВЕЩЕНИЯ)
CHANGE HISTORY (ИСТОРИЯ ИЗМЕНЕНИЙ)
STATUS (СТАТУС)
LINKS (ЛИНКИ)
OTHERS…

Priority & Severity

Серьезность (SEVERITY) — это атрибут, характеризующий влияние дефекта на работоспособность приложения.

Приоритет (PRIORITY) — это атрибут, указывающий на очередность выполнения задачи или устранения дефекта. Можно сказать, что это инструмент менеджера по планированию работ. Чем выше приоритет, тем быстрее нужно исправить дефект.

SEVERITY

S1 Блокирующая (Blocker)

Блокирующая ошибка, приводящая приложение в нерабочее состояние, в результате которого дальнейшая работа с тестируемой системой или ее ключевыми функциями становится невозможна. Решение проблемы необходимо для дальнейшего функционирования системы.

S2 Критическая (Critical)

Критическая ошибка, неправильно работающая ключевая бизнес логика, дыра в системе безопасности, проблема, приведшая к временному падению сервера или приводящая в нерабочее состояние некоторую часть системы, без возможности решения проблемы, используя другие входные точки. Решение проблемы необходимо для дальнейшей работы с ключевыми функциями тестируемой системой.

S3 Значительная (Major)

Значительная ошибка, часть основной бизнес логики работает некорректно. Ошибка не критична или есть возможность для работы с тестируемой функцией, используя другие входные точки.

S4 Незначительная (Minor)

Незначительная ошибка, не нарушающая бизнес логику тестируемой части приложения, очевидная проблема пользовательского интерфейса.

S5 Тривиальная (Trivial)

Тривиальная ошибка, не касающаяся бизнес логики приложения, плохо воспроизводимая проблема, малозаметная посредствам пользовательского интерфейса, проблема сторонних библиотек или сервисов, проблема, не оказывающая никакого влияния на общее качество продукта.

PRIORITY

P1 Высокий (High)

Ошибка должна быть исправлена как можно быстрее, т.к. ее наличие является критической для проекта.

P2 Средний (Medium)

Ошибка должна быть исправлена, ее наличие не является критичной, но требует обязательного решения.

P3 Низкий (Low)

Ошибка должна быть исправлена, ее наличие не является критичной, и не требует срочного решения.

Порядок исправления ошибок по их приоритетам:

High -> Medium -> Low

Требования к количеству открытых багов

Наличие открытых дефектов **P1, P2 и S1, S2**, считается **неприемлемым** для проекта. Все подобные ситуации требуют **срочного решения** и идут под контроль к менеджерам проекта.

Наличие строго **ограниченного количества открытых** ошибок **P3 и S3, S4, S5 не является критичным** для проекта и допускается в приложении. **Количество же открытых ошибок зависит** от размера проекта и установленных критериев качества.

Все требования к открытым ошибкам **оговариваются и документируются** на этапе принятия решения о качестве разрабатываемого продукта.

STATUS

New

Approved

Needs more info

Working

Solution Completed

Failed QA

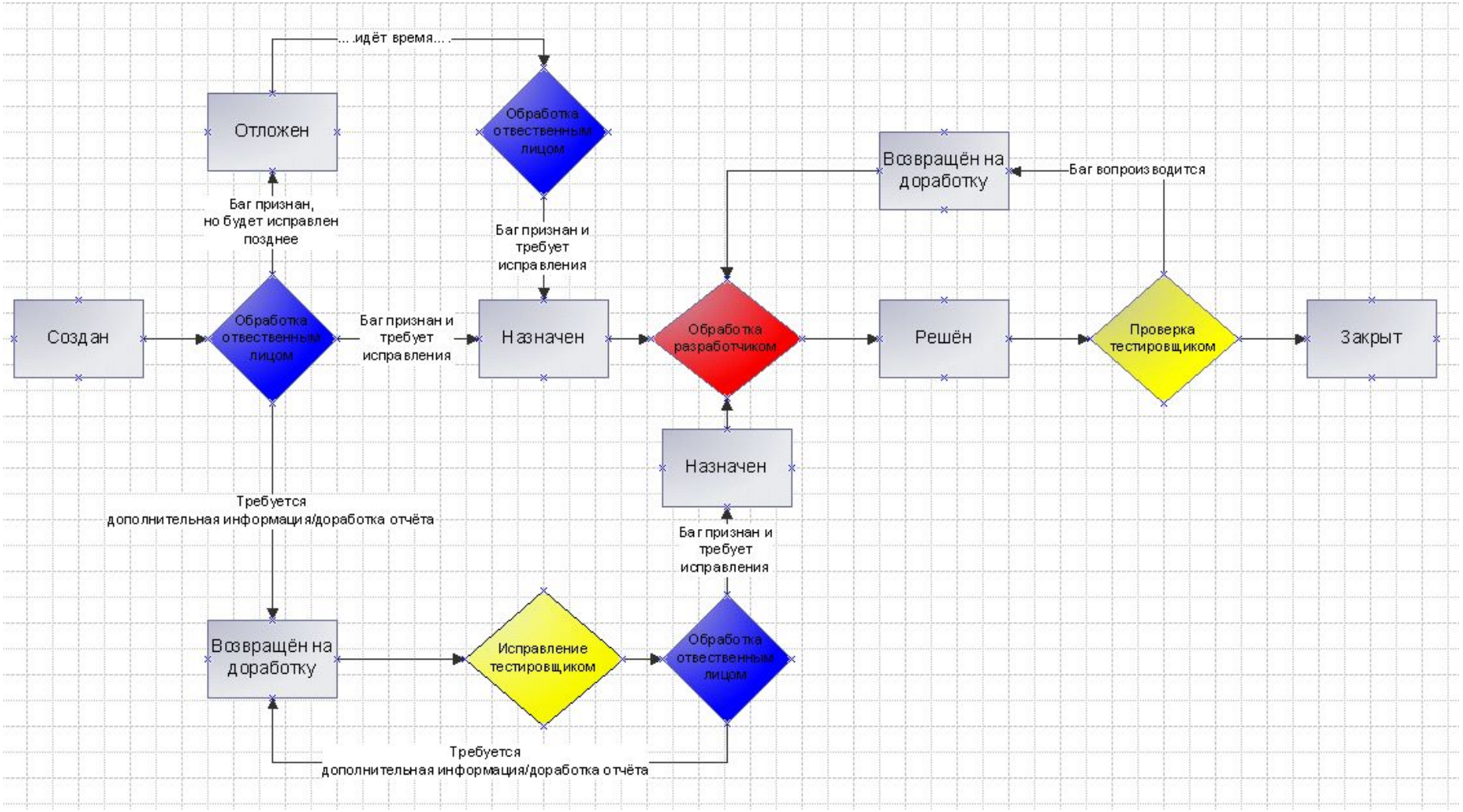
Passed QA

Tested

Deployed

Closed

ЖЦ Бага


... идёт время...
Отложен
Обработка ответственным лицом
Возвращён на доработку
Баг вопроизводится
Баг признан, но будет исправлен позднее
Баг признан и требует исправления
Создан
Обработка ответственным лицом
Баг признан и требует исправления
Назначен
Обработка разработчиком
Решён
Проверка тестировщиком
Закрыт
Назначен
Требуется дополнительная информация/доработка отчёта
Баг признан и требует исправления
Возвращён на доработку
Исправление тестировщиком
Обработка ответственным лицом
Требуется дополнительная информация/доработка отчёта

Test Plan

Тест план — это документ, описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения.

Минимум

Что надо тестировать?

описание объекта тестирования: системы, приложения, оборудования

Что будете тестировать?

список функций и описание системы и её компонент в отдельности

Как будете тестировать?

стратегия тестирования, а именно: виды тестирования и их применение по отношению к объекту тестирования

Когда будете тестировать?

последовательность проведения работ: подготовка (Test Preparation), тестирование (Testing), анализ результатов (Test Result Analisys) в разрезе запланированных фаз разработки

Критерии начала тестирования

- готовность тестовой платформы (тестового стенда)
- законченность разработки требуемого функционала
- наличие всей необходимой документации
- ...

Критерии окончания тестирования

- результаты тестирования удовлетворяют критериям качества продукта:
- требования к количеству открытых багов выполнены
- выдержка определенного периода без изменения исходного кода приложения Code Freeze (CF)
- выдержка определенного периода без открытия новых багов Zero Bug Bounce (ZBB)
- ...

Test Plan

TEST PLAN TEMPLATE — NAME OF THE PRODUCT
PREPARED BY — NAMES OF PREPARERS, DATE

1. Table of contents
2. Introductions
3. Objectives and tasks
4. Testing Strategy
5. Hardware requirements
6. Environment requirements
7. Resources
8. Project Milestones
9. Others…

Виды тест планов

- Master Plan or Master Test Plan
- Test Plan
- Product Acceptance Plan

Чек-лист/ Тест кейс

Чек-лист (check list) — это документ, описывающий что должно быть протестировано.
Тестовый случай (Test Case) — это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части.
Тест-комплект (test suite) – набор тест кейсов для проверки определенной функции или ее части.

Test case attributes

УНИКАЛЬНЫЙ ID (**ID**)
ПРИОРИТЕТ (**Test Case Priority**)
ИДЕЯ (**Idea**)
ПОДГОТОВИТЕЛЬНАЯ ЧАСТЬ (**Add Info**)
ИСТОРИЯ РЕДАКТИРОВАНИЯ (**Revision History**)
ШАГИ (**Steps**)
ОЖИДАЕМЫЙ РЕЗУЛЬТАТ (**Expected Result**)
ФАКТИЧЕСКИЙ РЕЗУЛЬТАТ (**Actual Result**)
ПРИЛОЖЕНИЯ (**Attachments**)
…Others

Процесс написания тест-кейса (**test case generation**).

Процесс исполнением тест-кейса (**test case execution**).

Каждый тест-кейс, исполнение которого завершено, дает нам **одно из двух**:

1. **Положительный** исход (**PASS**), если ФР равен ОР,
2. **Отрицательный** исход (**FAIL**), если ФР не равен ОР: найден **баг!**

Сколько ожидаемых результатов может быть в одном тест-кейсе?

Плохой стиль:

1. Зависимость тест-кейсов друг от друга.
2. Нечеткая формулировка шагов.
3. Нечеткая формулировка идеи и/или ожидаемого результата.

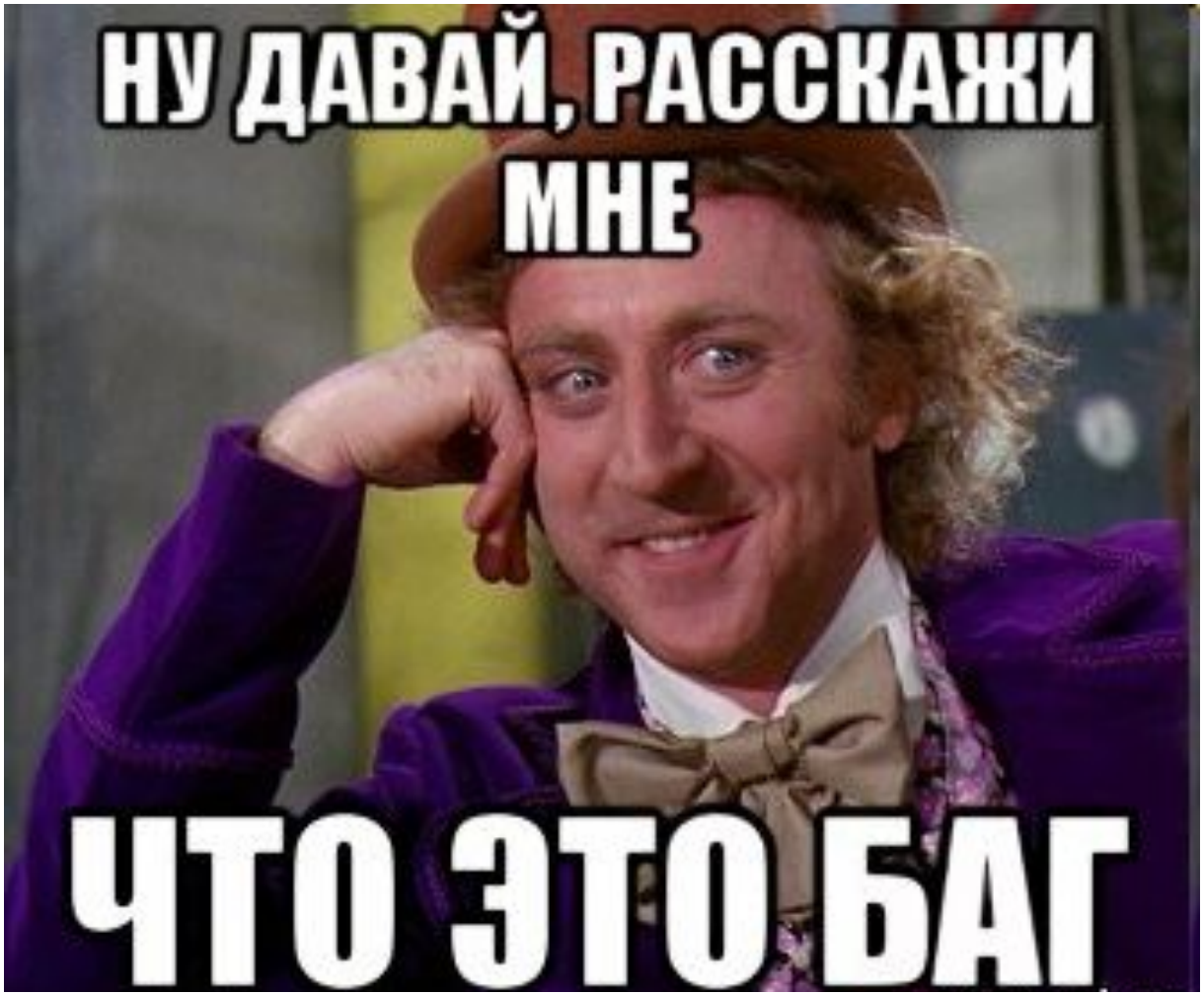
Testing report

Отчёт по тестированию — часть тестовой документации, включающая в себя описание процесса тестирования, суммарную информацию о протестированных за подотчётный период билдах, информацию о деятельности тестировщиков, а также некоторые статистические данные.

Testing report attributes

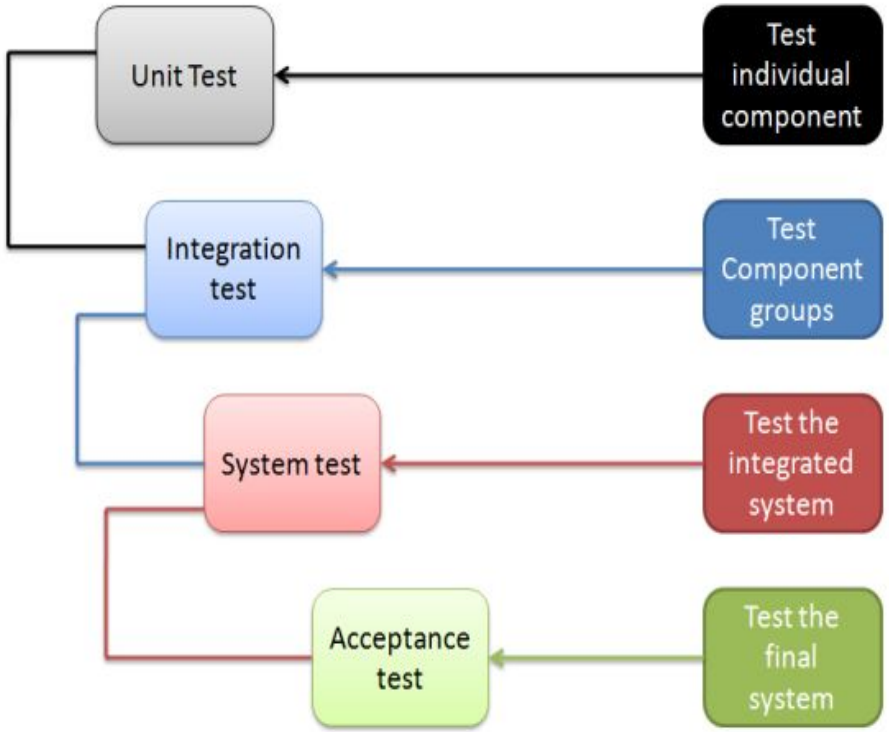
1. Builds number
2. List of test functions in the application
3. List of devices/coverage
4. List of defects
5. Blocker/Critical issues
6. Resources
7. Time Spent
8. Resume

Психология тестирования



Test Levels

1. Component testing (unit)
2. Integration testing
3. System testing
4. Acceptance testing



Testing types

1. По знанию внутренностей системы:

- черный ящик (black box testing);
- серый ящик (grey box testing);
- белый ящик (white box testing).

2. По объекту тестирования:

- функциональное тестирование (functional testing);
- тестирование интерфейса пользователя (UI testing);
- тестирование локализации (localization testing);
- тестирование скорости и надежности (load/stress/performance testing);
- тестирование безопасности (security testing);
- тестирование опыта пользователя (usability testing);
- тестирование установки (Installation testing);
- тестирование документации (documentation testing);
- тестирование взаимодействия (interoperability Testing);
- тестирование на отказ и восстановление (Failover and Recovery Testing);
- конфигурационное тестирование (Configuration Testing).

3. По субъекту тестирования:

- альфа-тестировщик (alpha tester);
- бета-тестировщик (beta tester).

4. По времени проведения тестирования:

до передачи пользователю — альфа-тестирование (alpha-testing);

- тест приемки (smoke test, sanity test или confidence test);
- тестирование новых функциональностей (new feature
- testing);
- регрессивное тестирование (regression testing);
- тест сдачи (acceptance or certification test);

после передачи пользователю — бета-тестирование (beta testing).

5. По критерию "позитивности" сценариев:

- позитивное тестирование (positive testing);
- негативное тестирование (negative testing).

6. По степени изолированности компонентов:

- компонентное тестирование (component testing);
- интеграционное тестирование (integration testing);
- системное тестирование (system).

7. По степени автоматизированности тестирования:

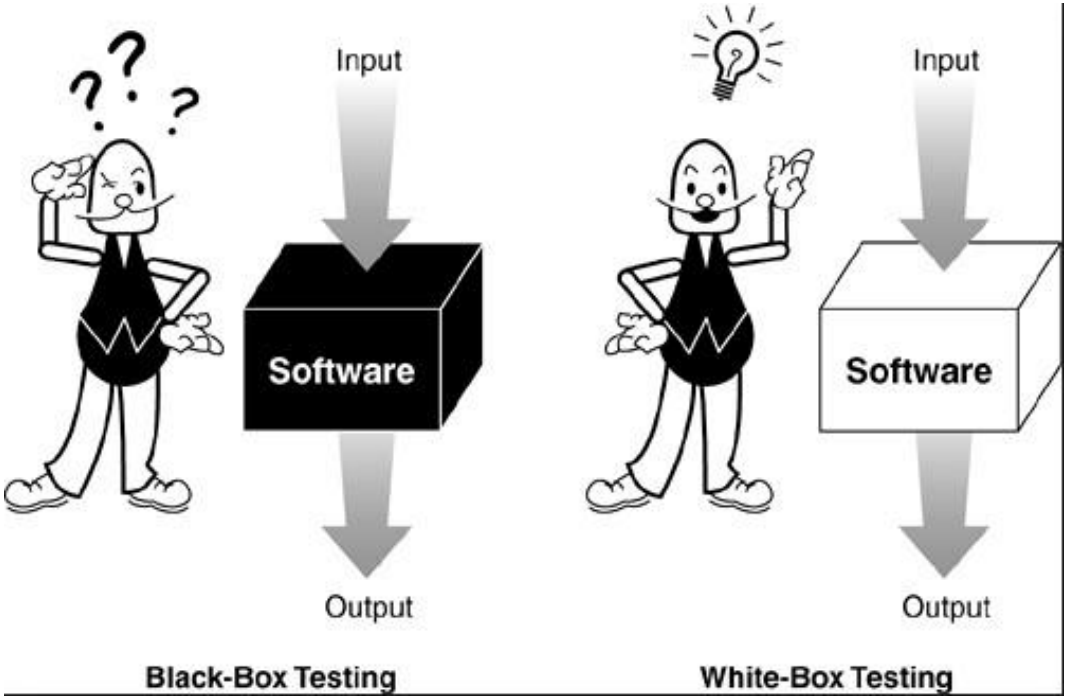
- ручное тестирование (manual testing);
- автоматизированное тестирование (automated testing);
- смешанное/полуавтоматизированное тестирование (semi automated testing).

8. По степени подготовки к тестированию:

- тестирование по документации (formal/documented testing);
- эд хок-тестирование (ad hoc testing).

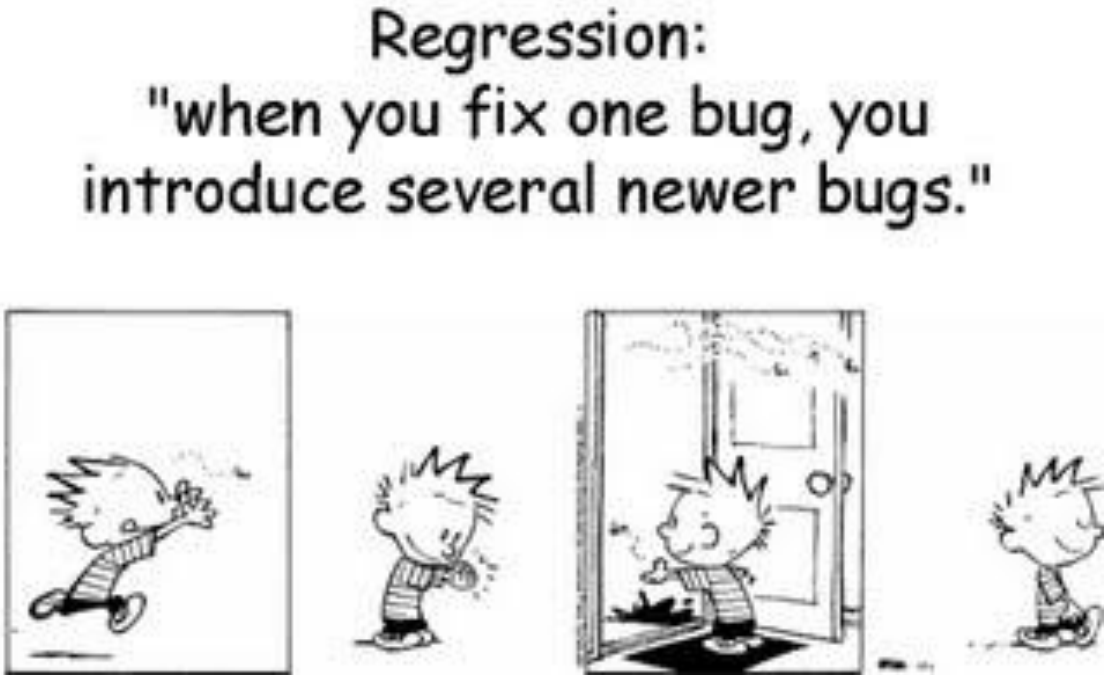
Structural testing

- “white box” or “glass box” or “clear-box testing”



Testing related to changes

- Re-testing
- Regression testing



Maintenance testing

Maintenance testing - testing that is provided after deployment called maintenance testing.

Maintainability testing - It basically defines that how easy it is to maintain the system. This means that how easy it is to analyze, change and test the application or product.

Литература:

1. http://www.protesting.ru/
2. http://ru.qahelp.net/
3. http://habrahabr.ru/
4. https://ru.wikipedia.org
5. The Scrum Master Training Manual, v. 1.2., By Nader K. Rad, Frank Turley, Copyright © 2013 Management Plaza.
6. «Тестирование Дот Ком или пособие по жесткому обращению с багами в интернет-стартапах» Р. Савин.

Homework

Максимально покрыть тест-кейсами автомат по выдаче жетонов в метро.