Виды и направления тестирования

Классификация по Куликову

1. По запуску кода на исполнение
   1. Статическое - без запуска кода

Например, тестирование требований, схем БД, графические прототипы, макеты и т.д. Или это м.б. код ревью, параметры среды окружения, тестовые данные и т.д.

* 1. Динамическое - с запуском кода

Приложение целиком или модули, часть кода

1. По доступу к коду и архитектуре приложения
   1. Метод белого ящика

Полный доступ к коду, достаточно квалификации для понимания, преимущественно тестируется разработчиком

* 1. Метод черного ящика

Без доступа к коду, на основе спецификаций (нет доступа или сознательно не обращается к коду для тестирования пользовательского пути, например, или взаимодействия других приложений)

* 1. Метод серого ящика

Частичный доступ к коду, комбинация белого и черного

1. По степени автоматизации
   1. Ручное (некрупный, временный проект, когда трудозатраты на автоматизацию будут дороже)
   2. Автоматизированное (CI/CD, крупный проект, нагрузочное)
2. По уровню детализации приложения (по уровню тестирования)
   1. Модульное (unit, преимущественно разработчиком, проверка блока кода. Например, блок для ввода имени в форме регистрации)
   2. Интеграционное (взаимодействие модулей)
   3. Регрессионное (оценка работоспособности уже проверенного кода после внесения в него изменений) (ПО с постоянными доработками)
   4. Системное (вся система целиком с точки зрения конечного пользователя)
   5. Приемочное (на соответствие ТЗ заказчика)
3. По степени важности тестируемых функций (по убыванию)
   1. СМОУК (ключевого функционала (в основном автотесты), sanity test (для более стабильной предрелизной версии))
   2. Тестирование критического пути (значимые элементы и функции приложения при стандартном использовании)
   3. Расширенное тестирование (вся остальная функциональность, расставленная по приоритетам, в т.ч. самые маловероятные случаи)
4. По принципам работы с приложением
   1. Позитивное (по спецификации, только корректные данные)
   2. Негативное (работа приложения с некорректными данными или операциями, потенциально приводящие к ошибкам)
5. По степени формализации (степени следованию документации)
   1. Тестирование на основе тест-кейсов (на основе заранее написанных тест-кейсов, тест-сьютов и другой документации)
   2. Исследовательское (по сценарию поведения, чек-листы)
   3. Свободное (интуитивное) Ad hoc (без документации, полагаясь на профессионализм и интуицию)

По целям (конечный желаемый результат) и задачам (проблемная ситуация, требующая разрешения с заданной целью):

1. Позитивное
2. Негативное
3. Функциональное (какие задачи для пользователя решает приложение)
4. Нефункциональное (удобство, совместимость, производительность, безопасность, надежность, расширяемость, требования...)
5. Регрессионное
6. Повторное
7. Инсталляционное (установка в новую срезу исполнения, апгрейд, даунгрейд, переинсталляция, повторный запуск инсталляции при прерванной ранее инсталляции, удаление, установка после удаления, установка нового приложения из семейства приложений, автоинсталляция)
8. Операционное (тестирование окружения: ОС, СУБД, серверы, аппаратное обеспечение и т.д.)
9. Интерфейса (корректная работа интерфейса)
10. Тестирование удобства пользования (юзабилити) (удобная работа интерфейса)
11. Доступности (спецвозможности)
12. Приёмочное (с точки зрения заказчика/конечного пользователя)
13. Интернационализации (возможность для адаптации локализации)
14. Локализации (качество адаптации под конкретную локализацию)
15. Безопасности (на сколько безопасно использование для пользователя, защищенность данных, может ли приложение противостоять злоумышленникам)
16. Тестирование БД (полнота, структурированность, непротиворечивость, целостность, адекватность)
17. Совместимости (с окружением, конфигурационное, кроссбраузерное, с мобильными аппаратами)
18. Использования ресурсов (эффективно ли приложение использует ресурсы)
19. Сравнительное
20. Демонстрационное
21. Надежности
22. Отказоустойчивости
23. Восстанавливаемости (производительность, данных и др)
24. Производительности (скорость реакции на внешние воздействия)
    1. Нагрузочное (сохраняет ли качество при нагрузке)
    2. Масштабируемость (сможет ли увеличить производительность если будет больше ресурсов)
    3. Объемность (при обработке больших объемов данных)
    4. Стрессовое (нагрузка значительно превосходящее (преуменьшающее) нежели при обычных нагрузках)
    5. Конкурентное (работа многопоточных приложений (несколько активных процессов, их взаимодействие))