

Объект тестирования: бар.

Вид тестирования	Краткое определение	Тестовые проверки
Functional Testing	Тестирование в целях проверки реализуемости функциональных требований, то есть способности объекта в определённых условиях решать задачи, нужные пользователям	Проверка на возможность реализации напитков покупателям. Корректировка графика работы
Safety Testing	Тестирование программного продукта с целью определить его способность при использовании оговоренным образом оставаться в рамках приемлемого риска причинения вреда здоровью, бизнесу, программам, собственности или окружающей среде	Проверка на безопасность использования объекта для покупателей в сильном опьянении. Исследовать алгоритм проверки покупателя на совершеннолетие в случае заказа алкогольного напитка.
Security Testing	Тестирование с целью оценить защищенность программного продукта от внешних воздействий. В нашем случае степень защищенности человека при эксплуатации системы	Тестирование защищенности объекта от экстремальных погодных условий и других внешних воздействий, способных деконструировать продукт. Попытки воздействовать на персонал физической силой, попытки украсть продукцию
Compatibility Testing	Проверка работоспособности объекта в различных условиях и средах (относительно ПО к примеру кроссплатформенное тестирование)	Проверка работоспособности объекта в отдельном здании, в арендованном помещении, в районах города с различным уровнем жизни, в глухом лесу

GUI Testing	Тестирование, выполняемое путем взаимодействия с системой через графический интерфейс пользователя	Тестирование заказа напитков через бармена, через другого посетителя, тестирование заказа напитка через приложение
Usability Testing	Тестирование с целью определения степени понятности, легкости в изучении и использовании, привлекательности программного продукта для пользователя при условии использования в заданных условиях	Предложить нескольким людям попробовать использовать объект: заказать напиток, выпить, попросить счет. Насколько быстро можно найти нужный напиток в меню, степень интуитивности работы системы.
Accessibility Testing	Тестирование, которое определяет степень легкости, с которой пользователи с ограниченными способностями могут использовать систему или ее компоненты	Проверка того, как люди с ограниченными способностями, могут попасть в бар и заказать напиток (только с их согласия)
Internationalization Testing	Тестирование адаптации продукта к языковым и культурным особенностям целого ряда регионов	Испытания на всех континентах
Performance Testing	Тестирование с целью определения производительности (нагрузочное, объемное, стрессовое)	Использование с разной интенсивностью, измерение скорости обслуживания посетителей. Предельное число осуществимых заказов, срок пригодности посуды, мебели, покрытия стойки
Stress Testing	Тестирование с определением надежности и устойчивости системы в условиях превышения пределов границ рабочих нагрузок	Заказываем одну кружку пива, заказываем -1 кружку пива, заказываем «jhbsjkb» кружек пива, спрашиваем бармена, где находится туалет

Negative Testing	Тестирование по нестандартному сценарию, которые соответствуют внештатному поведению тестируемой системы	Устроить драку, поджечь и разлить крепкий алкогольный напиток, напасть на бармена, попытаться выломать дверь
Black Box Testing	Тестирование системы без знания внутренней структуры и компонентов	Манипулируем составными частями, пытаемся понять как работает система
Automated Testing	Тестирование, позволяющее исключить человека из выполнения некоторых задач по оценке системы	Автоматическая регулировка освещения в зависимости от времени суток, установка автомата для автоматической оплаты заказов
Unit/Component Testing	Проверка работоспособности отдельных модулей системы	Тестирование бармена на стрессоустойчивость, тестирование посуды на прочность, тестирование напитков на соответствие заявленному составу
Integration Testing	Тестируется взаимодействие между отдельными модулями	Тестирование работоспособности объекта с изменением состояния компонентов продукта, к примеру тестирование возможности заказа напитков во время обслуживания другого посетителя

3. Разработать композицию тестов для первой поставки программного обеспечения (build 1), состоящей из трех модулей (модуль 1, модуль 2, модуль 3).

Build №1: Smoke + NFT_{АТ}.

Поверхностное тестирование (Smoke Test) выполняется для определения пригодности сборки для дальнейшего тестирования; полное тестирование системы или ее части как на корректных, так и на некорректных данных/сценариях (Acceptance Test, АТ) позволяет обнаружить дефекты и внести запись о них в багтрэкинг-систему.

4. Разработать композицию тестов для второй поставки программного обеспечения (build 2): исправлены заведенные дефекты, доставлена новая функциональность – модуль 4.

Build №2: Smoke + DV + (NFT_{AT}) + RT_{МАТ}.

Если не была добавлена новая функциональность, то DV + RT_{МАТ}. То есть, выполняется проверка исправления дефектов программистом (Defect Validation, DV), а также проверка работоспособности остальной функциональности после исправления дефектов на позитивных сценариях (Minimal Acceptance Test, МАТ).

Если была добавлена новая функциональность, то Smoke + DV + NFT_{AT} + RT_{МАТ}. В частности, выполняется поверхностное тестирование (Smoke Test), проверка исправления дефектов программистом (Defect Validation, DV), тестирование новых функциональностей (New Feature Testing, NFT), проверка старых функциональностей, т. е. регрессионное тестирование (Regression Test).

5. Разработать композицию тестов для третьей поставки программного обеспечения (build 3): заказчик решил расширять рынки сбыта и просит осуществить поддержку программного обеспечения на английском языке.

Build №3: Smoke + DV + Internationalization Testing.

Тестирование с поддержкой английского языка, проверка старых функциональностей, проверка исправленных багов.

6. Разработать композицию тестов для четвертой поставки программного обеспечения (build 4): заказчик хочет убедиться, что программное обеспечение выдержит нагрузку в 2000 пользователей.

Build №4: Smoke + DV + Performance Testing.

Тестирование производительности в 2000+ пользователей, автоматизация, оптимизация нагрузки.

Вывод: изучили классификацию видов тестирования, разработали проверки для различных видов тестирования, научились планировать тестовые активности в зависимости от особенностей поставляемой на тестирование функциональности.