Найпоширеніші типи тестування

Тестування на дим

Воно виявляє найбільш очевидні помилки, які можуть виникнути на перших етапах виконання програмного забезпечення. Тестування на дим займається збіркою програмного забезпечення.

Збірка — це «контейнер» багаторазових модулів, бібліотек, файлів даних, які реалізують функції програмного забезпечення. Інженери QA проводять тестування на дим, коли розробники кодують нові функції програмного забезпечення та інтегрують їх зі збірками. У цьому процесі розробники розгортають збірки програмного забезпечення в середовищі QA. Завдання тестувальника — забезпечити стабільність збірки; його перевірені функції працюють коректно, критичні функціональні можливості програмного забезпечення є бездоганними. Якщо тестування на дим пройшло успішно, команда QA може почати проводити більш серйозні тести.

Регресійне тестування

Як тільки розробники отримують готові частини програмного забезпечення, вони виправляють помилки та вносять необхідні зміни в код. Часто ці коригування є джерелами виникнення нових дефектів. А виявити їх допомагає регресійне тестування. Метою цього тестування є переконатися, що зміни програмного забезпечення не мали негативного впливу на існуючі функції. Регресійне тестування необхідне, коли: Виправлено проблеми з продуктивністю Програмне забезпечення містить нові функції Дефекти усунуто Вимоги були змінені Інженери QA зазвичай проводять серію з 3-5 сесій тестування. Хоча можна застосувати типи ручного тестування , це може бути дорогим і тривалим. Саме тому засоби автоматизації допомагають завершити регресійне тестування якомога швидше. Це заощаджує майже 60% часу на виявлення помилок і 40% витрат.

Функціональне тестування

Інженери QA перевіряють, що кожна функція програмного забезпечення працює відповідно до вимог. Це тестування не стосується вихідного коду програми. Швидше, він працює з результатами функціонування коду та відображає використання системи. Тестери надають кожній функції необхідні вхідні дані, перевіряють вихідні дані та порівнюють результати з очікуваними вимогами. Цей тест можна виконати вручну або за допомогою засобів автоматизації.

Інженери QA проводять функціональне тестування завдяки: Функціональним вимогам. Тут команда QA створює тестові випадки для процесу, які зосереджуються на попередніх функціях програмного забезпечення, які підлягають тестуванню. Інженери QA забезпечують коректну роботу функцій розробленого продукту з різними типами вхідних даних. Бізнес-процеси, які повинна забезпечувати програма. У цьому випадку тестувальники не перевіряють роботу окремих функцій програмного забезпечення. Вони забезпечують коректну роботу операцій за сценаріями використання системи. Таким чином, тестування базується на варіантах використання. Основною метою функціонального тестування є перевірка функціональності програмного забезпечення.

Зручність використання

Інженери QA проводять базове тестування зручності використання. Вони забезпечують користувачам вільну навігацію програмним забезпеченням за допомогою екранів і не стикаються з труднощами. Доступність: перевіряє, чи можуть користувачі легко розуміти і працювати з системою.

GUI

Графічний інтерфейс користувача (GUI) — це форма інтерфейсу, де ви взаємодієте з пристроями за допомогою зображень, піктограм та будь-яких візуальних елементів, які допомагають орієнтуватися в системі. Тестування графічного інтерфейсу перевіряє, чи правильно працює система інтерфейсу користувача продукту. Тестери також перевіряють, чи легко переміщатися додатком/програмним забезпеченням з точки зору користувачів. Відвідувачі веб-сайту залишають сайт, якщо їм не подобається інтерфейс або програма є складною для розуміння. Успіх програмного забезпечення залежить від того, як графічний інтерфейс взаємодіє з користувачами. Якщо відвідувачі сайту легко користуються різними функціями, це великий крок уперед до випуску високоякісного продукту. Ось чому тестувальники переконуються, що графічний інтерфейс не містить помилок.
 Ось контрольний список тестування GUI: Екрани з елементами керування значками, меню, зображеннями, кнопками.
Тестери переконаються, що ці елементи мають належний розмір, ширину, довжину, положення, чіткість. Усі типи панелей (вікна, діалогові вікна, панель інструментів) Шрифт, який використовується в програмі/додатку, має бути читабельним Розташування елементів графічного інтерфейсу правильне з різною роздільною здатністю екрана.

Тестування локалізації

Культурні моделі та мова є областями інтересу для тестування локалізації. Воно налаштовує програмне забезпечення/програму для цільової країни та її мовні особливості. У процесі локалізаційного тестування інженери QA повинні: перевірити помилки вмісту перевірити відповідність інтерфейсу користувача культурним особливостям перевірити, чи призначені мова за замовчуванням, формат дати й часу, валюта відповідно до цільової країни Таким чином, тестувальники гарантують, що ваша цільова аудиторія почувається комфортно на вашому сайті, оскільки він поводиться відповідно до місцевих культурних шаблонів.

Інтеграційне тестування

Зазвичай програмний проект складається з програмних модулів, створених командою розробників. Часто один розробник кодує один модуль з індивідуальними характеристиками. Програміст може випадково поміняти параметри функції місцями. В результаті під час виконання виклику виникає помилка, і кілька модулів не працюють разом. Після того, як окремі елементи програми готові, спеціалісти QA приступають до тестування їх спільного функціонування. Інтеграційне тестування перевіряє єдине функціонування модулів і забезпечує їх безперебійну роботу. Під час тесту інженери QA спочатку об’єднують модулі в пару, а потім об’єднують їх у великі блоки. Нарешті, елементи повинні бути об’єднані в єдину систему.
Команда QA проводить інтеграційне тестування, коли:
Модулі взаємодіють з API та іншими інструментами сторонніх розробників. Тестери перевіряють, що API правильно приймає дані модуля.
Вимоги змінені. Іноді розробники впроваджують нові зміни, які не проходять модульне тестування. У таких випадках інтеграція стає необхідністю

Тестування на сумісність

Сучасні програми та веб-сайти повинні ідеально працювати з різними веб-браузерами, обладнанням, мобільними пристроями та операційними системами. Завдання тестування на сумісність полягає в аналізі поведінки системи в різних середовищах.
Тест проводиться двома методами:
Переднє тестування перевіряє сумісність програми з найновішими мобільними операційними системами
Зворотне тестування гарантує, що програмне забезпечення, створене для останніх версій середовища, також працює зі старими.
Іншими словами, поведінка системи має бути повністю сумісною з будь-якою

версією апаратного чи програмного забезпечення. Існує кілька найбільш поширених помилок, які виявляє тестер сумісності: Проблеми з вирівнюванням тексту та розміром шрифту Зламаний інтерфейс користувача, смуга прокрутки, таблиці, рамки Зміни в CSS, кольорі та стилі Тестування запобігає фінансовим ризикам та пошкодженню репутації через погану сумісність системи та негативний досвід клієнтів.

Тестування продуктивності

Функціональність програмного забезпечення не є єдиною проблемою для інженерів QA. Час відгуку, масштабованість, надійність також відіграють вирішальну роль. Метою тестування продуктивності є вимірювання часу відповіді програмного забезпечення при очікуваному робочому навантаженні. Це запобігає повільному завантаженню програмного забезпечення при одночасному входженні кількох відвідувачів, непостійній роботі в різних операційних системах і поганому використанні.
Контрольний список тестування продуктивності:
Стабільність: забезпечення стабільності програмного забезпечення при змінних умовах навантаження
Масштабованість: визначення максимального навантаження на користувача, яке може впоратися з програмним забезпеченням
Швидкість: щоб перевірити, чи швидка відповідь системи

Стрес-тестування

Тест аналізує поведінку системи в умовах високого трафіку та обробки даних протягом тривалого періоду часу. За умов тривалого, постійного та екстремального навантаження система відображає відповідне повідомлення про помилку.

Стрес-тестування гарантує, що програмне забезпечення відновлюється після збою та продовжує функціонувати. Фахівці з QA проводять стрес-тестування паралельно з навантажувальним тестуванням, поступово збільшуючи навантаження на сервер.
Після досягнення навантаження тестери аналізують і вимірюють час відповіді протягом вибраного періоду часу.
Стрес-тестування корисно в таких ситуаціях:

Веб-сайт відчуває раптовий сплеск реклами у провідних ЗМІ Інтернет-магазини стикаються зі сплесками відвідуваності під час святкових сезонів і періодів знижок Тест запобігає великій втраті доходу через збій системи. Ваш веб-сайт краще підготовлений до нестандартних умов і забезпечує позитивний досвід клієнтів. Часто веб-сайти стикаються з раптовими сплесками після запуску рекламних або маркетингових кампаній для залучення трафіку.

Таким чином, навантажувальне тестування визначає, чи справляється програмне забезпечення зі збільшенням трафіку на сервері.
Інженери QA фокусуються на:

- Операційна потужність програмного забезпечення
- Стабільність веб-сайту з піковим навантаженням трафіку
- Максимальна кількість користувачів для входу на сайт

Приймально-здавальне випробування

Це завершальний етап процесу тестування програмного забезпечення. Тест оцінює, чи відповідає система вимогам бізнесу та чи готова до випуску. Команда QA також проводить його щодо потреб користувачів. Тестування приймання визначає, чи задоволена цільова аудиторія використанням програмного забезпечення.
Завдяки цьому етапу тестування клієнти завжди мають чітке розуміння специфіки програмного забезпечення. Крім того, власники продуктів мають можливість зробити сайт повністю відповідним потребам клієнтів.

Тестування API

Нещодавній перехід до архітектур на основі API є потужним драйвером інновацій у індустрії програмного забезпечення. Веб та інтерфейс користувача більше не створюють критичних ризиків для бізнесу. Натомість нелюдський інтерфейс API став набагато вразливішим. З цієї причини тестування API має стати важливою частиною стратегії якості продукту.
Інтерфейс прикладного програмування (API) є важливою частиною будь-якої програми. API містять функції, які виконує інша система. Таким чином, API є фоном для обміну даними між різними частинами програмного забезпечення.
А завдання забезпечення якості API — перевірити, чи правильно функціонують програмні системи. Тестери визначають, чи працюють API відповідно до вимог безпеки, функціональності, продуктивності та надійності. Тестер API зосереджується на функціональності основного програмного забезпечення. Тестування API дозволяє отримати доступ до програми без інтерфейсу користувача. Таким чином інженери QA виявляють невеликі помилки, які можуть викликати серйозні проблеми на етапі тестування графічного інтерфейсу. Тестери перевіряють, чи працюють API взагалі, чи легко вони інтегруються в інші системи, а також їхню продуктивність у стресових умовах. На відміну від тестування графічного інтерфейсу, забезпечення якості API займає менше часу. Майже 3000 тестів API займають 50 хвилин, на стільки ж GUI-тестів потрібно 30 годин. Інструменти автоматизації для тестування API вимагають менше коду і, отже, прискорюють процес QA.