**Тестирование как способ обеспечения качества программного обеспечения.**

**Тестирование** — это процесс проверки программного обеспечения, системы или приложения на соответствие определённым требованиям и оценки их качества.

Качество программного продукта характеризуется набором свойств, определяющих, насколько продукт "хорош" с точки зрения заинтересованных сторон, таких как заказчик продукта, спонсор, конечный пользователь, разработчики и тестировщики продукта, инженеры поддержки, сотрудники отделов маркетинга, обучения и продаж.

**Цель тестирования** — выявить ошибки, неполадки и нежелательное поведение программного продукта. Чем раньше обнаруживается дефект, тем дешевле его исправление и тем меньше последствий он несёт как для компании-разработчика, так и для непосредственного пользователя продукта.

**Существует несколько видов тестирования программного обеспечения:**

Модульное тестирование: Проверка отдельных модулей или компонентов программы для подтверждения их корректной работы.

Интеграционное тестирование: Тестирование взаимодействия между несколькими модулями или компонентами для выявления дефектов, возникающих при их интеграции.

Системное тестирование: Проверка всей системы в целом для обеспечения ее соответствия требованиям и спецификациям.

Приемочное тестирование: Оценка системы с точки зрения пользователя для подтверждения ее готовности к эксплуатации.

Регрессионное тестирование: Проверка того, что изменения в программном обеспечении не привели к появлению новых дефектов в уже работающих функциях.

Нагрузочное тестирование: Оценка производительности системы при высокой нагрузке для выявления ее пределов.

**Виды тестирования:**

**Статическое тестирование**. Проверка кода и документации без запуска приложения или программы.

**Динамическое тестирование**. Проводится с запуском ПО. Тестировщики оценивают, долго ли грузятся страницы, сколько оперативной памяти нужно для нормальной работы приложения.

**Функциональное тестирование**. Проверяет, справляется ли приложение с возложенными на него функциями и задачами.

**Нефункциональное тестирование**. Исследует совместимость и производительность компонентов приложения.

**Тестирование «чёрного ящика»** Тестировщик работает только с интерфейсной частью продукта и не видит его код.

**Тестирование «белого ящика»** Инженер видит код проекта.

**Тестирование «серого ящика»** Тестирование, при котором тестировщик имеет частичный доступ к исходному коду или документации.

Тестирование играет ключевую роль в обеспечении качества программного обеспечения. Это процесс оценки программного обеспечения с целью обнаружения дефектов и несоответствий требованиям. По сути, это контролируемый способ поиска проблем до того, как они попадут к конечным пользователям.

**Почему тестирование важно для обеспечения качества:**

Обнаружение дефектов: Главная задача тестирования - найти ошибки, баги, уязвимости и другие дефекты в коде. Чем раньше обнаружены эти проблемы, тем дешевле и проще их исправить.

Проверка соответствия требованиям: Тестирование подтверждает, что программное обеспечение работает так, как задумано и соответствует определенным требованиям и спецификациям.

Повышение надежности и стабильности: Обнаружив и исправив дефекты, тестирование делает программное обеспечение более надежным и стабильным, что снижает вероятность сбоев и ошибок в процессе эксплуатации.

Улучшение пользовательского опыта: Тестирование помогает убедиться, что программное обеспечение удобно в использовании, интуитивно понятно и соответствует ожиданиям пользователей.

Снижение рисков: Обнаружение и устранение дефектов до релиза снижает риски, связанные с негативными отзывами пользователей, финансовыми потерями, репутационными убытками и даже юридическими последствиями.

Обеспечение безопасности: Тестирование помогает выявить уязвимости в системе безопасности, которые могут быть использованы злоумышленниками для получения несанкционированного доступа или нанесения вреда.

Увеличение доверия: Успешно протестированное и надежное программное обеспечение вызывает больше доверия у пользователей и заинтересованных сторон.

**Различные виды тестирования:**

**Существует множество видов тестирования, каждый из которых направлен на проверку различных аспектов программного обеспечения. Некоторые из наиболее распространенных:**

Модульное тестирование (Unit Testing): Тестирование отдельных компонентов (модулей) программного обеспечения.

Интеграционное тестирование (Integration Testing): Тестирование взаимодействия между различными модулями или компонентами.

Системное тестирование (System Testing): Тестирование всего программного обеспечения как единого целого.

Приемочное тестирование (Acceptance Testing): Тестирование программного обеспечения конечными пользователями для проверки соответствия их требованиям.

Регрессионное тестирование (Regression Testing): Повторное тестирование после внесения изменений в код для убеждения, что новые изменения не привели к появлению новых дефектов.

Производительное тестирование (Performance Testing): Оценка производительности программного обеспечения под различными нагрузками.

Тестирование безопасности (Security Testing): Проверка программного обеспечения на наличие уязвимостей и обеспечение его безопасности.

Функциональное тестирование (Functional Testing): Проверка того, что программное обеспечение выполняет свои функции в соответствии со спецификациями.

Нефункциональное тестирование (Non-Functional Testing): Оценка таких характеристик программного обеспечения, как производительность, надежность, удобство использования и безопасность.

UI/UX тестирование (UI/UX Testing): Проверка удобства и визуальной привлекательности пользовательского интерфейса.

Альфа-тестирование (Alpha Testing): Внутреннее тестирование продукта перед выпуском.

Бета-тестирование (Beta Testing): Тестирование продукта реальными пользователями вне команды разработчиков.

**Стратегии тестирования:**

**Выбор подходящей стратегии тестирования зависит от конкретных потребностей и целей проекта. Вот некоторые общие стратегии:**

Водопадная модель (Waterfall Model): Тестирование выполняется после завершения каждого этапа разработки.

Agile-разработка: Тестирование выполняется итеративно и параллельно с разработкой.

TDD (Test-Driven Development): Разработка начинается с написания тестов, а затем пишется код, который проходит эти тесты.

BDD (Behavior-Driven Development): Тесты пишутся на основе поведения системы с точки зрения пользователя.

**Инструменты тестирования:**

Существует множество инструментов тестирования, которые помогают автоматизировать и упростить процесс тестирования. Они включают в себя:

Инструменты для модульного тестирования: JUnit, NUnit, pytest.

Инструменты для автоматизированного тестирования UI: Selenium, Cypress, Playwright.

Инструменты для тестирования производительности: JMeter, LoadRunner.

Инструменты для управления тестированием: TestRail, Zephyr, qTest.

\_\_\_\_\_

Концепция тестирования - это общая стратегия и подход к тестированию программного обеспечения, которая определяет, каким образом будет проводиться тестирование, какие задачи оно должно решать и какие результаты оно должно получить.

Тестирование является ключевым этапом в разработке программного обеспечения и способом обеспечения его качества. Правильно спланированное и проведенное тестирование помогает выявить и устранить ошибки и проблемы до выпуска программы, что повышает доверие пользователей к продукту.

**Разработать модульные тесты для программного модуля AsyncAdd, содержащий метод асинхронного чтения всех строк из файла. Код для программного модуля находится «Resorce/ModuleC#/». Оформить тест-кейсы для модульного тестирования.**