

Методы и средства тестирования информационных систем

Функциональное тестирование

Функциональное тестирование – это тип тестирования программного обеспечения, при котором система тестируется на соответствие функциональным требованиям/спецификациям.

Каждая функция работает в соответствии с функциональными требованиями.

Функциональное тестирование в первую очередь включает в себя тестирование «черного ящика» и не касается исходного кода приложения.

Ключевые причины важности функционального тестирования:

1. Проверка функциональности
2. Выявление дефектов и ошибок
3. Повышение качества программного обеспечения
4. Обеспечение совместимости
5. Достижение удовлетворенности клиентов

Этапы функционального тестирования:

1. Определить и проанализировать, какую функциональность необходимо протестировать
2. Написать тест-кейсы
3. Подготовка тестовых данных
4. Проведение тестирования
5. Составление отчета по результатам тестирования

Функциональное тестирование может включать в себя следующие **виды тестирования**:

1. Модульное тестирование
2. Интеграционное тестирование
3. Системное тестирование
4. Регрессионное тестирование
5. Санитарное тестирование
6. Дымовое тестирование

Модульное (unit) тестирование – это тип тестирования, при котором тестируются отдельные модули или компоненты ПО.

Выполняется разработчиками на этапе кодирования

Единица для измерения – функция, метод, объект, процедура...

Интеграционное тестирование – это тип тестирования, при котором программные модули объединяются логически и тестируются как группа.

Выявляются баги при взаимодействии модулей (в первую очередь проверка обмена данными)

Системное тестирование – это тип тестирования, при котором оценивается работа системы как единого целого с целью определения качества выполнения системных требований.

Оно проверяет дизайн и поведение системы, часто выходя за пределы, прописанные в требованиях

Обычно выполняется QA-командой, независимой от разработчиков, что помогает сохранять объективность

Системное тестирование может быть как функциональным, так и нефункциональным

Приемочное тестирование – это тип тестирования, при котором система проверяется на приемлемость – готовность к передаче заказчику.

Проверка соответствия требованиям пользователей и бизнес-требованиям, проверка соответствия критериям приемки

После завершения приемочного тестирования пользователи/заказчики решают, принимать ли систему в пользование

Модульное



Интеграционн
о
е



Системное



Приемочное

В рамках функционального тестирования программного продукта проверяются следующие **компоненты** и **критерии**:

пользовательский интерфейс

API

базы данных

безопасность систем

работа клиентских и серверных приложений

общая функциональность

Инструменты для ручного функционального тестирования

Системы управления тестированием:

TestIT – это система управления тестированием, которая была создана тестировщиками для тестировщиков. С её помощью удобно хранить тест-кейсы, составлять тест-планы, создавать прогоны и управлять ими.

TestRail – удобно создавать чек-листы, тест-кейсы и прогоны, выгружать результаты прогонов, отчеты о тестировании и сами тест-кейсы в формат CSV. Также сравнивать результаты нескольких прогонов. Он поддерживает интеграцию с различными баг-трекинг-системами (Jira, YouTrack и т.д.).

Allure – с его помощью удобно управлять ручным и автоматизированным тестированием. Легко разрабатывать тест-кейсы и чек-листы, создавать тестовые прогоны и собирать статистику по результатам тестирования.

Инструменты для работы с БД:

DBeaver – универсальный инструмент для управления БД с удобным интерфейсом. С помощью него можно работать с различными СУБД: MySQL, PostgreSQL, SQLite, Oracle и другими.

SQL Developer – графический интерфейс для работы с БД и выполнения SQL-запросов. С его помощью можно создавать и выполнять запросы, исследовать базы данных и отслеживать ошибки.

HeidiSQL – графическая оболочка для работы с MySQL, PostgreSQL и Microsoft SQL Server. Она может быть использована для создания, изменения и удаления таблиц, вставки и удаления данных, создания SQL-запросов и многого другого.

Инструменты для тестирования API:

Postman – с его помощью можно составлять и отправлять запросы, собирать коллекции и делиться ими с коллегами. Также в Postman можно писать автотесты для тестирования API.

SoapUI – с помощью данного инструмента можно легко и удобно тестировать как SOAP, так и REST-сервисы. Можно проверять работоспособность веб-сервисов, устанавливать доступность, работу различных запросов и отслеживать получение ответов.

Swagger UI – инструмент для описания и проверки API-методов. К каждому запросу есть пример ответа и описание приходящих в них параметров. Не требует установки на устройство пользователя.

Методы функционального тестирования:

1. Положительное тестирование
2. Отрицательное тестирование
3. Проверки на основе конечного пользователя
4. Тесты эквивалентности
5. Граничное тестирование
6. Тесты на основе принятия решений
7. Альтернативные проверки кода
8. Специальные тесты

Д/З: Описать инструменты для автоматизации функционального тестирования