Лекция на тему "Разработка плана тестирования системы. Построение тест-кейсов и сценариев тестирования" охватывает основные принципы и процессы, связанные с тестированием программного обеспечения. Тестирование является неотъемлемой частью жизненного цикла разработки ПО и помогает обеспечить качество системы, её соответствие требованиям и отсутствие критических ошибок. Планирование тестирования, создание тест-кейсов и сценариев тестирования играют ключевую роль в данном процессе.

**1. Введение в тестирование систем**

**Тестирование системы** — это процесс проверки соответствия программы требованиям, а также выявления ошибок, дефектов или уязвимостей в системе. Основная цель тестирования — убедиться, что система функционирует корректно, безопасно и эффективно при различных условиях использования.

**Типы тестирования:**

* **Функциональное тестирование:** Проверка того, что система выполняет все функции в соответствии с требованиями.
* **Нефункциональное тестирование:** Проверка аспектов производительности, безопасности, удобства использования и других нефункциональных характеристик.
* **Ручное тестирование:** Тестировщик вручную выполняет сценарии и проверяет поведение системы.
* **Автоматизированное тестирование:** Использование инструментов для автоматического выполнения тестов и проверки результатов.

**2. Разработка плана тестирования системы**

**План тестирования** — это документ, описывающий стратегию тестирования, задачи, объем тестирования, ресурсы, график и риски. Основная задача плана тестирования — предоставить четкие и организованные шаги по выполнению всех тестов системы.

**Основные элементы плана тестирования:**

1. **Цели тестирования:** Определение целей и задач тестирования. Например, проверка основных функциональностей, проверка интеграции между модулями или тестирование безопасности системы.
2. **Объем тестирования:** Описание того, что будет тестироваться, а что исключено из объема тестирования. Это помогает сфокусировать усилия на ключевых частях системы. Например, можно определить, что тестирование включает веб-приложение, но не затрагивает мобильное приложение.
3. **Типы тестирования:** Указание видов тестирования, которые будут использованы (функциональное, нагрузочное, интеграционное и т.д.).
4. **Методология тестирования:** Определение подходов, которые будут использоваться для выполнения тестирования — например, мануальное или автоматизированное тестирование, использование методологий TDD (Test-Driven Development) или BDD (Behavior-Driven Development).
5. **Ресурсы:** Описание ресурсов, необходимых для проведения тестирования — это могут быть сотрудники, оборудование, программные инструменты.
6. **График тестирования:** Определение сроков выполнения различных этапов тестирования, включая дедлайны и зависимость от других процессов разработки.
7. **Критерии приемки:** Описание критериев, при которых система будет считаться успешно прошедшей тестирование. Например, процент успешных тестов или допустимый уровень дефектов.
8. **Риски и предположения:** Определение возможных рисков (например, нехватка времени или оборудования) и предположений (например, наличие стабильной версии ПО для тестирования).

**3. Построение тест-кейсов**

**Тест-кейсы** (test cases) — это конкретные сценарии, которые описывают шаги для проверки определенных функций системы и ожидаемые результаты. Тест-кейсы являются основой для систематического тестирования и документирования процесса.

**Основные элементы тест-кейса:**

1. **Название тест-кейса:** Краткое описание того, что проверяет тест. Например, "Тест авторизации пользователя с правильными учетными данными".
2. **Описание:** Описание цели теста и контекста, в котором он используется. Например, "Этот тест проверяет, что пользователь с правильными учетными данными может успешно войти в систему".
3. **Предусловия:** Условия или данные, которые должны быть выполнены до начала теста. Например, "Пользователь должен быть зарегистрирован в системе".
4. **Шаги выполнения:** Подробное описание шагов, которые необходимо выполнить для прохождения теста. Пример:
   1. Открыть страницу входа.
   2. Ввести логин и пароль.
   3. Нажать кнопку "Войти".
5. **Ожидаемый результат:** Описание того, что должно произойти, если система работает корректно. Например, "Пользователь успешно авторизован и перенаправлен на главную страницу".
6. **Фактический результат:** Записывается после выполнения теста — фактический результат системы, который сравнивается с ожидаемым.
7. **Статус теста:** Указывается после выполнения — успешный тест или проваленный.

**Пример тест-кейса:**

**Тест-кейс 001: Авторизация пользователя с корректными данными**

* **Описание:** Проверка, что пользователь может авторизоваться в системе с корректными учетными данными.
* **Предусловия:** Пользователь зарегистрирован в системе.
* **Шаги:**
  1. Перейти на страницу входа.
  2. Ввести логин "user@example.com" и пароль "password123".
  3. Нажать "Войти".
* **Ожидаемый результат:** Пользователь авторизован и перенаправлен на главную страницу.
* **Фактический результат:** (заполняется после выполнения теста).
* **Статус:** Успешно/Неудачно.

**4. Построение сценариев тестирования**

**Сценарии тестирования** (test scenarios) — это более высокоуровневые тесты, которые описывают последовательность действий пользователя или системы. В отличие от тест-кейсов, сценарии могут охватывать несколько функциональностей или проверять сложные пользовательские потоки. Они часто используются для интеграционного или системного тестирования.

**Пример сценария тестирования:**

**Сценарий 1: Регистрация нового пользователя и создание заказа**

1. Открыть главную страницу сайта.
2. Перейти на страницу регистрации.
3. Заполнить форму регистрации (имя, электронная почта, пароль).
4. Подтвердить регистрацию.
5. Авторизоваться в системе с использованием только что созданных учетных данных.
6. Выбрать товар в каталоге и добавить его в корзину.
7. Перейти к оформлению заказа.
8. Ввести платежные данные и подтвердить заказ.
9. Убедиться, что заказ успешно оформлен и отображается в личном кабинете пользователя.

**5. Категории тест-кейсов и сценариев тестирования**

1. **Позитивные тесты:** Проверяют, что система работает правильно при вводе корректных данных и выполнении правильных действий. Например, тест авторизации с правильными учетными данными.
2. **Негативные тесты:** Проверяют поведение системы при вводе некорректных данных или выполнении недопустимых действий. Например, тест авторизации с неверными учетными данными или тест на попытку создания заказа без указания адреса доставки.
3. **Регрессионные тесты:** Эти тесты выполняются после внесения изменений в систему (например, исправления багов или добавления новых функций) для проверки того, что старые функции не нарушены.
4. **Интеграционные тесты:** Эти тесты проверяют, как различные модули системы взаимодействуют между собой. Например, как взаимодействуют модули авторизации и корзины покупок.
5. **Нагрузочные тесты:** Проверяют, как система работает под высокой нагрузкой. Например, сценарий тестирования может заключаться в том, чтобы 1000 пользователей одновременно создали заказы.

**6. Инструменты для создания и выполнения тест-кейсов**

Для управления тест-кейсами и сценариями тестирования часто используются специализированные инструменты. Некоторые из них:

* **JIRA + Zephyr:** Управление задачами разработки и тестирования, включая создание и отслеживание тест-кейсов.
* **TestRail:** Платформа для планирования и выполнения тестов, с возможностью создания отчетов.
* **Selenium:** Инструмент для автоматизированного тестирования веб-приложений.
* **JUnit/TestNG:** Фреймворки для автоматизированного тестирования Java-приложений.

**7. Валидация тест-кейсов**

После создания тест-кейсов они должны быть валидированы. Это включает проверку следующих аспектов:

* **Полнота:** Все требования системы должны быть покрыты соответствующими тестами.
* **Корректность:** Тесты должны правильно проверять требуемую функциональность и содержать правильные шаги.
* **Исполнимость:** Тесты должны быть выполнимы в текущей версии системы и давать однозначные результаты.

**8. Заключение**

Планирование и выполнение тестирования системы является ключевым этапом в обеспечении её качества. Разработка тест-кейсов и сценариев тестирования позволяет структурировать процесс тестирования, сделать его систематическим и повторяемым. Использование правильных методологий и инструментов позволяет ускорить процесс тестирования, повысить его эффективность и минимизировать количество ошибок в системе.