# Самостоятельная работа по теме: Тестовый сценарий и тестовый пакет

**Выполнил студент 319 группы: Меркушин Евгений**

### **1. Теоретическая часть:**

1. **Тестовый сценарий** — это подробное пошаговое руководство для выполнения теста, включающее описание условий, шагов и ожидаемых результатов. Он описывает, как будет проводиться тестирование определенного функционала или компонента программного обеспечения.
2. **Тестовый пакет** — набор тестовых сценариев, объединенных для комплексного тестирования определенной функциональности или модуля программного обеспечения. Пакет может включать различные типы тестов (позитивные, негативные, нефункциональные и т.д.) и служит для систематического подхода к проверке работоспособности системы.
3. **Позитивный тест** — тест, направленный на проверку корректной работы системы при нормальных условиях, когда вводимые данные соответствуют ожидаемым значениям. Его цель — убедиться, что функционал работает правильно при стандартных сценариях использования.
4. **Негативный тест** — тест, предназначенный для проверки реакции системы на некорректные или непредусмотренные условия, например, при вводе недопустимых значений или выполнении запрещённых действий. Этот тип теста помогает выявить слабые места и ошибки в обработке исключительных ситуаций.
5. **Нефункциональное тестирование** — вид тестирования, ориентированный на проверку характеристик программного обеспечения, не связанных напрямую с его функциональностью. Сюда входят такие аспекты, как производительность, масштабируемость, удобство использования, безопасность и совместимость.

# Пример тестового сценария на тему "Регистрация нового пользователя":

**Идентификатор:** TS-001**Название:** Регистрация нового пользователя**Описание:** Проверяет функциональность регистрации нового пользователя на сайте.**Предусловия:**

* Сайт доступен и работает корректно.
* Поле для ввода email и пароля отображается на странице регистрации.

**Шаги выполнения:**

1. Открыть страницу регистрации.
2. Ввести валидный email в поле "Email".
3. Ввести сложный пароль в поле "Пароль", соответствующий требованиям сайта (например, минимум 8 символов, одна заглавная буква, одна цифра и специальный символ).
4. Повторить ввод пароля в поле подтверждения пароля.
5. Нажать кнопку "Зарегистрироваться".

**Ожидаемый результат:**

* Появляется сообщение об успешной регистрации.
* Пользователь автоматически входит в свою учётную запись.
* На указанный email приходит письмо с подтверждением регистрации.

# Структура тестового пакета для модуля "Корзина покупок" интернет-магазина:

**Тестовый пакет: Корзина покупок**

1. **Добавление товара в корзину:**
   * Название: Добавление одного товара в пустую корзину.
   * Ожидаемый результат: Товар успешно добавляется в корзину, сумма корзины обновляется.
2. **Удаление товара из корзины:**
   * Название: Удаление единственного товара из заполненной корзины.
   * Ожидаемый результат: Корзина становится пустой, сумма пересчитана и равна нулю.
3. **Редактирование количества товаров в корзине:**
   * Название: Изменение количества товара в корзине.
   * Ожидаемый результат: Количество товара изменяется, общая сумма пересчитается и отображена корректно.

### **2. Практическая часть:**

Для тестирования раздела 'Профиль пользователя' в социальном приложении можно использовать следующие виды тестов:

# 1. Проверка редактирования профиля (изменения имени и фамилии, настройки личных данных)

**Виды тестов:**

* **Функциональное тестирование** — проверка корректности работы функций редактирования различных полей профиля (имя, фамилия, дата рождения, город проживания и т.д.). Например, нужно убедиться, что при изменении имени оно сохраняется и отображается корректно.
* **Юзабилити-тестирование** — оценка удобства интерфейса для пользователя при внесении изменений в профиль. Важно, чтобы кнопки были интуитивно понятны, поля легко заполнялись, а уведомления об успехе или ошибках были четкими и информативными.
* **Интерфейсовое тестирование** (UI-тестирование) — проверка того, что элементы интерфейса (кнопки, текстовые поля, выпадающие списки и т.п.) работают должным образом и выглядят одинаково на разных устройствах и браузерах.
* **Тестирование на локализацию** — если приложение поддерживает разные языки, нужно протестировать, как изменяются тексты и интерфейсные элементы при переключении языка.

**Что будет проверяться:**

* Возможность изменения имени и фамилии.
* Сохранение внесенных изменений.
* Валидность введенных данных (например, нельзя ввести цифры вместо букв в полях имени и фамилии).
* Работа уведомлений о сохранении изменений.
* Удобство навигации и заполнения формы.

# 2. Проверка загрузки аватара (поддержка форматов файлов, проверка на ограничения по размеру)

**Виды тестов:**

* **Функциональное тестирование** — проверка возможности загрузки изображений различных поддерживаемых форматов (JPG, PNG, GIF и т.д.) и соблюдение ограничений по размеру файла.
* **Негативные тесты** — попытка загрузки неподдерживаемых форматов и файлов слишком большого размера для проверки сообщений об ошибках и корректной обработки таких случаев.
* **Производительность и нагрузка** — тестирование скорости загрузки больших файлов и поведения системы при массовой загрузке аватаров несколькими пользователями одновременно.
* **Безопасность** — проверка на возможность загрузки вредоносных файлов или вставки злонамеренного кода через загрузку изображений.

**Что будет проверяться:**

* Поддерживаемые форматы изображений (JPG, PNG и другие).
* Ограничения по максимальному размеру файла (например, не больше 5 MB).
* Сообщения об ошибке при попытке загрузки неподдерживаемого формата или превышенном размере файла.
* Скорость загрузки и сохранение изображения в профиле.
* Безопасность процесса загрузки (отсутствие возможности внедрения вредоносного кода).

# 3. Проверка обработки ошибок при загрузке изображения неподдерживаемого формата

**Виды тестов:**

* **Негативное тестирование** — специально пытаемся загрузить файл неподдерживаемого типа (например, DOC, PDF, EXE) для проверки реакции системы.
* **Сообщения об ошибке** — важно убедиться, что при возникновении ошибки пользователь получает понятное и полезное сообщение, объясняющее проблему.
* **Логирование ошибок** — нужно проверить, что все ошибки записываются в журнал событий для дальнейшего анализа и исправления.

**Что будет проверываться:**

* Реакция системы на попытку загрузки неподдерживаемого файла.
* Корректность сообщения об ошибке для пользователя.
* Запись ошибки в лог-файл для дальнейшей диагностики.
* Отсутствие зависаний или сбоев приложения при возникновении такой ошибки.

**3. Дополнительное задание:**

Отслеживание покрытия тестами играет ключевую роль в обеспечении высокого уровня качества программного обеспечения. Покрытие тестами показывает, какая часть кода была протестирована, и помогает определить, насколько полно проверено поведение программы. Вот несколько причин, почему важно следить за покрытием тестами:

# Причины важности отслеживания покрытия тестами:

1. **Выявление непроверенных участков кода:**
   * Высокий процент покрытия тестами указывает на то, что большая часть кода подверглась проверке. Однако даже небольшие участки кода, оставшиеся вне зоны тестирования, могут содержать серьезные ошибки, которые могут проявиться позже.
2. **Повышение уверенности в стабильности системы:**
   * Чем выше покрытие тестами, тем меньше вероятность появления неожиданных ошибок при использовании программы. Это дает уверенность в том, что большинство потенциальных проблем было обнаружено и исправлено на этапе тестирования.
3. **Оптимизация процесса разработки:**
   * Когда команда знает, какие части кода недостаточно покрыты тестами, она может сконцентрироваться на создании дополнительных тестов для этих областей. Это помогает распределить усилия и ресурсы более эффективно.
4. **Документирование и отчетность:**
   * Отчет о покрытии тестами предоставляет наглядную информацию о состоянии тестирования. Это полезно для менеджеров проектов, заказчиков и других заинтересованных сторон, чтобы оценить степень готовности продукта к выпуску.

# Примеры влияния недостаточного покрытия на качество ПО:

1. **Скрытые баги:**
   * Если некоторые важные функции или ветви кода остаются непокрытыми тестами, существует высокая вероятность того, что там скрываются ошибки. Эти баги могут остаться незамеченными до момента реального использования программы, что приведет к сбоям или другим нежелательным последствиям.
2. **Проблемы с безопасностью:**
   * Недостаточное покрытие тестами может оставить уязвимости в системе, которые хакеры смогут использовать для атаки. Особенно это актуально для приложений, работающих с конфиденциальными данными.
3. **Неожиданные изменения в поведении системы:**
   * Непротестированные участки кода могут вести себя непредсказуемо при изменениях в окружении или новых версиях операционной системы. Это может привести к нестабильной работе программы и ухудшению пользовательского опыта.
4. **Трудоемкость поддержки и доработок:**
   * Программное обеспечение с низким уровнем покрытия тестами сложнее поддерживать и дорабатывать. Любые изменения могут привести к появлению новых ошибок, так как нет уверенности в том, что они не нарушают существующие функциональные возможности.

# Пример из реальной практики:

Представьте себе банковское мобильное приложение, в котором функция перевода средств покрыта тестами лишь частично. Некоторые редкие сценарии, такие как перевод очень крупных сумм или переводы между счетами в разных валютах, остались без внимания. В результате пользователи начинают сталкиваться с проблемами при совершении подобных транзакций, что вызывает недовольство клиентов и репутационные потери для банка.

Таким образом, отслеживание покрытия тестами помогает не только выявить недостатки в тестировании, но и предотвратить потенциальные проблемы, которые могли бы возникнуть в будущем. Это важный аспект обеспечения качества программного обеспечения и снижения риска выхода на рынок продукта с серьезными дефектами.

**4. Аналитическая часть:**

Недостаточное тестирование функций аутентификации и авторизации может привести к серьёзным рискам для безопасности и надёжности программного обеспечения. Рассмотрим основные проблемы, которые могут возникнуть в этом случае:

# 1. Утечка данных и компрометация аккаунтов

Если функции аутентификации не полностью протестированы, злоумышленники могут воспользоваться уязвимостями для получения несанкционированного доступа к личным данным пользователей. Это может привести к краже паролей, номеров кредитных карт, медицинских записей и другой чувствительной информации.

**Пример:** Неправильная реализация механизма хранения паролей (например, хранение паролей в открытом виде вместо их хеширования) может позволить злоумышленникам получить доступ ко всей базе данных пользователей.

# 2. Бреши в безопасности

Недостаточные тесты могут пропустить уязвимости, такие как SQL-инъекции, XSS-атаки или CSRF (Cross-Site Request Forgery). Эти уязвимости дают возможность злоумышленнику выполнять произвольные запросы к базе данных, внедрять вредоносный код или изменять состояние сессии пользователя.

**Пример:** Если механизм аутентификации подвержен SQL-инъекциям, злоумышленник может обойти процедуру входа в систему, выполнив запрос, который позволит ему войти в аккаунт другого пользователя.

# 3. Нарушение прав доступа

Неполное тестирование функций авторизации может привести к ситуации, когда пользователи получают доступ к ресурсам, которым они не должны иметь доступ. Это нарушает принципы контроля доступа и может привести к утечке конфиденциальной информации или повреждению данных.

**Пример:** Некорректная настройка ролей и привилегий может дать обычным пользователям права администратора, позволяя им удалять или изменять важные данные.

# 4. Потеря доверия пользователей

Когда пользователи сталкиваются с проблемами безопасности, такими как взлом аккаунтов или утечка данных, они теряют доверие к приложению или услуге. Это приводит к оттоку пользователей, негативным отзывам и возможным судебным разбирательствам.

**Пример:** Крупные утечки данных, вызванные слабой системой аутентификации, могут нанести значительный ущерб репутации компании и привести к потере клиентов.

# 5. Финансовые потери

Компании могут столкнуться с финансовыми убытками из-за штрафов, судебных издержек и компенсации ущерба пострадавшим пользователям. Кроме того, потеря клиентской базы и снижение доходов также могут быть результатом недостаточного тестирования функций аутентификации и авторизации.

**Пример:** В случае утечки платежных данных, компания может быть обязана выплатить компенсацию пострадавшим клиентам, а также оплатить услуги по расследованию инцидента и устранению последствий.

# 6. Неэффективная работа системы

Некоторые ошибки в функциях аутентификации могут приводить к сбоям в работе системы, таким как бесконечные циклы аутентификации или невозможность входа в систему для легитимных пользователей. Эти проблемы снижают доступность сервиса и ухудшают пользовательский опыт.

**Пример:** Ошибка в механизме одноразовых паролей может привести к тому, что пользователи не смогут получить доступ к своим аккаунтам, что вызовет массовую блокировку и необходимость ручного разблокирования.

# Подведение итогов:

Недостаточное тестирование функций аутентификации и авторизации представляет собой серьезный риск для любого программного обеспечения. Оно может привести к утрате данных, нарушению безопасности, финансовым потерям и снижению доверия пользователей. Для минимизации этих рисков необходимо проводить всестороннее тестирование, охватывающее все аспекты аутентификации и авторизации, включая обработку ошибок, безопасность данных и контроль доступа.

### **5. Заключение:**

Тестовые сценарии и тестовые пакеты играют ключевую роль в процессе тестирования программного обеспечения, обеспечивая структурированную и систематическую проверку его функциональности и надежности.

**Тестовые сценарии** представляют собой подробные инструкции для выполнения конкретных проверок, описывающие шаги, условия и ожидаемые результаты. Они помогают тестировщикам точно следовать плану тестирования, гарантируют полноту охвата всех важных аспектов функционала и облегчают воспроизводимость тестов. Благодаря этому повышается эффективность тестирования и снижается вероятность пропуска критических ошибок.

**Тестовые пакеты** объединяют наборы тестовых сценариев, направленных на комплексную проверку определенных модулей или функциональностей ПО. Они обеспечивают целостный подход к тестированию, охватывая различные аспекты, такие как позитивные и негативные тесты, функциональные и нефункциональные тесты. Тестовые пакеты способствуют упорядоченному выполнению тестов, упрощают управление процессом тестирования и предоставляют возможность автоматизации повторяемых задач.

Таким образом, тестовые сценарии и тестовые пакеты являются необходимыми инструментами для обеспечения качества программного обеспечения. Их правильное составление и использование позволяют выявить и устранить ошибки на ранних этапах разработки, минимизировать риски, связанные с выпуском некачественного продукта, и повысить общую надежность и стабильность системы.