### **Кейс: Система для отдела тестирования с использованием ИИ**

#### **Цели разработки системы:**

1. **Автоматизация создания тестов**:
   * Генерация тест-кейсов на основе предоставленных материалов (описаний, скриншотов, видео, записи действий пользователя).
   * Автоматическое создание автотестов для популярных фреймворков (Selenium, Playwright или их аналоги).
2. **Оптимизация процесса тестирования**:
   * Уменьшение времени на разработку тестов.
   * Обеспечение более полного покрытия функциональности за счет анализа исходных данных.
3. **Динамическая актуализация тестов**:
   * Использование данных из Swagger-документации, исходного кода backend (Symfony) и frontend (React) для понимания связей между функциональностью и тестами.
   * Автоматическое обновление зависимых тестов при изменении кода.
4. **Поддержка высококачественного тестирования**:
   * Устранение человеческого фактора на этапе создания и обновления тестов.
   * Своевременное уведомление о возможных изменениях в тестах с предложением ревью.

#### **Основные функции системы:**

1. **Генерация тест-кейсов**:
   * На основе описания функциональности, скриншотов, видео взаимодействия с системой или записей действий в специальном ПО.
   * Автоматическое создание текстовых сценариев с шагами выполнения.
   * Интеграция с существующими системами управления тест-кейсами (Jira, TestRail и др.).
2. **Генерация автотестов**:
   * Создание готовых к использованию автотестов для Selenium, Playwright или их аналогов.
   * Возможность выбора фреймворка для автотестов.
   * Генерация тестов с комментариями для облегчения ревью.
3. **Анализ Swagger-документации API**:
   * Автоматическое создание тестов на основе эндпоинтов, описанных в Swagger.
   * Определение тестовых сценариев для проверки валидных и невалидных данных.
   * Интеграция с backend (Symfony) для анализа кода и построения зависимостей.
4. **Анализ исходного кода**:
   * Поддержка кода backend (Symfony) и frontend (React).
   * Анализ изменений в коде и их влияния на существующие тесты.
   * Автоматическое обновление тестов с учетом изменений в логике и структуре кода.
5. **Ревью и обновление тестов**:
   * Генерация отчетов о предлагаемых изменениях в тестах.
   * Возможность предварительного просмотра и корректировки тестов перед их внедрением.
   * Уведомления о конфликтующих тестах и рекомендация по их устранению.
6. **Обучение модели на основе обратной связи**:
   * Система обучается на правках и дополнениях тестов.
   * Со временем вырабатывает стиль и предпочтения отдела тестирования.

#### **Возможности интеграции:**

1. **Системы управления тест-кейсами**:
   * Jira, TestRail, Xray.
2. **CI/CD-пайплайны**:
   * Jenkins, GitLab CI/CD, GitHub Actions.
3. **Среды разработки**:
   * Поддержка IDE (VS Code, PhpStorm, WebStorm) для удобной работы с автотестами.
4. **Источники данных для анализа**:
   * Swagger, Postman Collections.
   * Исходный код на Git.

#### **Роли в системе:**

1. **Тестировщик**:
   * Загрузка материалов для генерации тестов (скриншоты, видео, описания).
   * Ревью и подтверждение предлагаемых тест-кейсов и автотестов.
2. **Разработчик**:
   * Предоставление информации о коде (через Git-репозитории).
   * Анализ изменений, влияющих на тесты.
3. **Администратор**:
   * Настройка параметров интеграции (Swagger, Git, CI/CD).
   * Управление доступами и ролями пользователей.

#### **Ключевая задача команды:**

Разработать прототип системы с функционалом:

* Генерации тест-кейсов и автотестов на основе описаний, скриншотов, видео или записей действий.
* Интеграции с Swagger, Git и CI/CD для анализа и поддержки актуальности тестов.
* Автоматического обновления зависимых тестов при изменении кода.
* Ревью и управления тестами через удобный интерфейс.  
    
    
  **Контакты для вопросов:**   
  Технические вопросы: Владимир, @vdaron, 89604660100  
  Общие вопросы: Яна, @wild\_ambition, 89281348604