## **ЛР-1. Обзор и Сравнительный анализ трех информационных систем**

***Вводное к теме.*** В данный момент на коммерческих проектах часто вводится автоматизация тестирования. Бизнесу выгодно доверить информационной системе действия, которые вручную выполняются специалистами по тестированию ПО. В связи с этим остро стоит вопрос по выбору инструмента для автоматизации тестирования.

Перед рассмотрением аналогов важно определить, какие категории фреймворков существуют на рынке. Определение категорий позволит подобрать наиболее подходящий для проекта вариант. Особое внимание уделялось таким параметрам, как мультиплатформенность и поддержка языков программирования, сложность использования, особые функциональные возможности, популярность и совместимость.

***О наличии на рынке аналогов.*** На рынке существует большое количество фреймворков по автоматизации тестирования, каждый из них глобально выполняет одну и ту же задачу – автоматизировать ручные процессы при тестировании, но аналоги уникальны и имеют как свои преимущества, так и недостатки.

Каждый фреймворк нацелен на конкретную сферу тестирования. Некоторые созданы для автоматизации тестирования пользовательского интерфейса веб-приложений, некоторые оптимизированы для работы с настольными приложениями, некоторые имеют функционал для симуляции действий на мобильных устройствах, другие могут тестировать Application Programming Interface (API) или базы данных.

В результате информационного поиска, были выделены 6 основных категорий, описание каждой из них, их предназначение и примеры инструментов автоматизации, относящихся к этим категориям.

*Инструменты для веб-тестирования*

Эти инструменты предназначены для автоматизации тестирования веб-приложений, симулируют взаимодействие пользователя с веб-элементами страницы. Наиболее популярными и востребованными являются:

* *Selenium:* Наиболее популярный фреймворк для веб-автоматизации;
* *Cypress:* Современный инструмент с уникальными возможностями, работающий напрямую с версткой веб-приложения;
* *WebDriverIO*: Node.js-фреймворк для веб-тестирования;

*Инструменты для мобильного тестирования*

Эти инструменты предназначены для автоматизации тестирования мобильных приложений, взаимодействие пользователя с мобильным устройством. Например:

* *Appium:* Кросс-платформенный инструмент для мобильной автоматизации;
* *XCUITest* (для iOS) и *Espresso* (для Android);

*Инструменты для API-тестирования*

Эти инструменты помогают автоматизировать тестирование API для клиент-серверной и других архитектур. Примеры:

* *Postman:* Инструмент для создания и отправки HTTP-запросов;
* *RestAssured:* Java-библиотека для тестирования REST API;
* *RestSharp:* C#-библиотека для тестирования REST API;

*Инструменты для нагрузочного тестирования*

Эти инструменты позволяют оценить производительность системы под нагрузкой. Примеры:

* *JMeter:* Самый популярный инструмент для нагрузочного тестирования;
* *Gatling:* Scala-инструмент для тестирования производительности;

*Инструменты для тестирования на уровне базы данных*

* Эти инструменты помогают автоматизировать проверку данных в базах данных. Примеры:
* *DbUnit:* Java-библиотека для тестирования баз данных;
* *SQLUnit:* Инструмент для запуска SQL-запросов в тестах;

*Инструменты для функционального тестирования*

* Эти инструменты помогают автоматизировать функциональное тестирование. Примеры:
* *NUnit:* Фреймворк для тестирования на C#;
* *JUnit и TestNG:* Популярные фреймворки для тестирования на Java;

***Категории аналогов.*** Для определения категории, необходимой для сравнительного анализа – был проведён детальный анализ актуальности и популярности на современных проектах каждой из категорий.

В последние время набирают популярность zero-code решения, которые не требуют для своего использования знаний языков программирования, достаточно просто записать действия и иметь возможность воспроизводить их в качестве автоматизированного теста. Примером является платформа TestGrid (Рисунок 1).



Рисунок 1. Интерфейс TestGrid

Но такие решения не будут рассматриваться для сравнения, так как решения, поддерживающие языки программирования более стабильны и эффективны для решения задач по тестированию.

По результатам анализа и опроса был сделан вывод, что самой популярной и важной на данный момент категорией является категория “Инструменты для веб-тестирования”. Далее предоставлено краткое обоснование выбора этой категории.

Веб-приложения продолжают оставаться востребованными и актуальными в современном мире. Вот несколько ключевых аспектов, подчеркивающих их популярность:

1. Рост числа веб-приложений. Каждый день появляется все больше веб-приложений – от интернет-магазинов до социальных сетей. Это связано с ростом онлайн-бизнеса, цифровой трансформацией и повышенным спросом на онлайн-сервисы;
2. Множество устройств и браузеров: Пользователи заходят на веб-сайты с разных устройств (компьютеры, смартфоны, планшеты) и используют разные браузеры. Веб-приложения должны быть адаптированы для всех сценариев;
3. Безопасность и защита данных: Веб-приложения подвергаются угрозам безопасности, таким как взломы, утечки данных и мошенничество. Тестирование и обеспечение безопасности – важные аспекты веб-разработки;
4. Постоянные обновления и новые требования: Веб-приложения постоянно обновляются, добавляются новые функции и исправляются ошибки. Разработчики должны следить за последними тенденциями и адаптироваться к новым требованиям;
5. Пользовательский опыт: Качество веб-приложения напрямую влияет на удовлетворенность пользователей. Отзывчивость, интуитивный интерфейс и быстрая загрузка – ключевые факторы успеха;

***Выбор исследуемых аналогов.*** После определения точной категории был составлен список самых часто используемых на практике инструментов автоматизации, входящих в эту категорию, которые выполняют свои задачи на современных проектах по разработке ПО и широко используются в сфере тестирования пользовательских интерфейсов и внутренней логики веб-приложений.

Список основных инструментов для автоматизации тестирования интерфейса веб-приложений, начиная от самого популярного к менее популярному:

* Selenium;
* Cypress;
* Playwright;
* Katalon Studio;
* TestCafe;
* WebDriverIO;
* Ranorex;
* Jest;
* Puppeteer;
* SikuliX;
* Applitools Eyes;
* UFT;
* Robot Framework;
* Screenster;
* CasperJS;
* Watir;
* Ghost Inspector;
* TestCraft;
* Cucumber;
* NightwatchJS;
* TestRigor;
* Rapise;
* Aquality framework;
* Testim;
* Subject7;
* Selenide;
* ACCELQ;
* iMacros;
* IBM Rational Functional Tester;
* Watir;
* Telerik Test Studio;
* Virtuoso;

Для сравнительного анализа необходимо определить тройку наиболее подходящих инструментов. Чтобы выбрать наилучшие фреймворки из списка были определены самые влиятельные критерии выбора.

Основные критерии для выбора исследуемых аналогов могут включать следующие аспекты:

* *Функциональность:* Какие функции и возможности предоставляют системы. Особое внимание уделялось на автоматизацию тестов, интеграцию с другими инструментами, отчетность и уникальные особенности инструмента.
* *Возможности:* Оценивается, насколько фреймворк соответствует потребностям проекта. Рассмотрена поддержка различных типов тестов (функциональные, нагрузочные, UI-тесты), интеграция с другими инструментами и возможности взаимодействия с элементами страницы.
* *Производительность:* Как быстро системы выполняют тесты. Есть ли ограничения по объему тестов.
* *Поддержка и сообщество:* Каков уровень поддержки со стороны разработчиков. Есть ли активное сообщество пользователей.
* *Сложность использования:* Насколько легко развернуть и настроить систему. Требуется ли специальная экспертиза.
* *Язык программирования:* Нужно убелиться, что фреймворк поддерживает язык программирования, востребованный для проекта.
* *Отчетность и логирование:*Какие отчеты генерируются после выполнения тестов. Хорошая отчетность поможет анализировать результаты.
* *Совместимость с инфраструктурой:*Как фреймворк интегрируется с существующей инфраструктурой (CI/CD-пайплайны, контейнеры и т. д.).
* *Бюджет и лицензия:*Некоторые фреймворки могут быть платными или иметь ограничения по использованию. Фреймворк должен соответствовать проектному бюджету и требованиям.

***Поочередный обзор аналогов.*** На основе выделенных критериев были определены три наиболее влиятельных фреймворка для сравнения: Selenium WebDriver, Cypress и Playwright.

Разбор фреймворка Selenium WebDriver представлен в таблице 1.

|  |  |
| --- | --- |
| **Критерий** | **Описание** |
| **Название и разработчик** | Selenium WebDriver. Разработан командой разработчиков и сообществом, без единого авторства. |
| **Описание и назначение** | Это инструмент для автоматизации тестирования веб-приложений и симуляции действий с браузером.  Он позволяет создавать и запускать тесты, взаимодействуя с веб-страницами и веб-элементами как это делает пользователь.  Особенностью является открытый исходный код фреймворка и бесплатное использование. |
| **Основные компоненты** | *Selenium WebDriver.* Это основной компонент, который позволяет взаимодействовать с браузерами. Есть возможность написать тесты на разных языках программирования (Java, Python, C#, Ruby или JavaScript) и выполнять их с помощью WebDriver.  *Selenium IDE*. Это расширение для браузера Chrome и Firefox, которое позволяет записывать и воспроизводить действия пользователя на веб-страницах.  *Selenium Grid*. Этот компонент позволяет распределить тесты на несколько машин, чтобы ускорить выполнение. |
| **Производительность** | Selenium обеспечивает эффективное выполнение тестов, но производительность может зависеть от объема тестов и настроек. |
| **Поддержка и сообщество** | Selenium имеет активное сообщество разработчиков и пользователей, что обеспечивает поддержку и решение проблем.  Также он является устоявшимся инструментом с обширной документацией и большим количеством примеров использования. |
| **Сложность использования** | Настройка Selenium может быть сложной, особенно для тестирования в разных браузерах и для запуска тестов в параллельных потоках. |
| **Язык программирования** | Поддерживает широкий спектр языков программирования: Java, Python, C#, Ruby, JavaScript. |
| **Отчетность и логирование** | Selenium предоставляет возможности для генерации отчетов после выполнения тестов.  Также имеет большое количество плагинов для генерации отчетов. |
| **Совместимость с инфраструктурой** | Selenium интегрируется с CI/CD-пайплайнами, контейнерами и другими инструментами. |
| **Бюджет и лицензия** | Selenium – это бесплатный и открытый инструмент, доступный для всех. С открытым исходным кодом |

Таблица 1. Описание Selenium WebDriver

Разбор фреймворка Cypress представлен в таблице 2.

|  |  |
| --- | --- |
| **Критерий** | **Описание** |
| **Название и разработчик** | Cypress разрабатывается Cypress.io, некоммерческой организацией, которая создает инструменты для тестирования веб-приложений. |
| **Функциональность и возможности** | Cypress – это инструмент для автоматизации веб-приложений, специализирующийся на тестировании фронтенда. Он обеспечивает простоту в создании и запуске тестов, а также взаимодействие с веб-страницами.  Особенностью является простой синтаксис и интегрированный тестовый раннер. |
| **Производительность** | Cypress обладает высокой производительностью благодаря своей архитектуре и способности выполнять тесты непосредственно в браузере. |
| **Поддержка и сообщество** | У Cypress есть активное сообщество, которое предоставляет поддержку, обучающие материалы и решения проблем. |
| **Сложность использования** | Cypress обладает простым и интуитивным синтаксисом, что упрощает его использование даже для новичков. |
| **Язык программирования** | Cypress поддерживает только JavaScript, что делает его доступным для менее широкого круга разработчиков. |
| **Отчетность и логирование** | Cypress предоставляет гибкие возможности для создания отчетов и логирования результатов тестов. |
| **Совместимость с инфраструктурой** | Cypress интегрируется с CI/CD-пайплайнами, Docker, и другими инструментами. |
| **Бюджет и лицензия** | Cypress – это бесплатный и открытый инструмент, доступный для всех. |

Таблица 2. Описание Cypress

Разбор фреймворка Playwright практически аналогичен разбору Cypress, поэтому выделены отличительные особенности в таблице 3. А также проанализированы преимущества и недостатки в таблице 4.

|  |  |
| --- | --- |
| **Критерий** | **Описание** |
| **Функциональность и возможности** | Playwright – это библиотека для автоматизации веб-приложений, специализирующаяся на тестировании фронтенда. |
| **Отличительные особенности** | Она обладает кросс-платформенной, кросс-браузерной и кросс-языковой поддержкой, а также включает полезные функции, такие как автоматическое ожидание.  Она специально разработана для современного веба и обычно выполняет тесты очень быстро, даже для сложных проектов тестирования |

Таблица 3. Описание особенностей Playwright

|  |  |
| --- | --- |
| **Преимущества** | **Недостатки** |
| **Кросс-браузерная поддержка**: Playwright работает с несколькими браузерами, включая Chromium (Chrome, Edge), Firefox и WebKit (Safari). Эта совместимость с разными браузерами позволяет проводить последовательное тестирование в разных средах. | **Сложность использования**: Настройка Playwright может быть сложной, требует документации и опыта работы с подобными системами |
| **Кросс-платформенная поддержка**: Вы можете использовать Playwright для тестирования приложений на разных платформах, таких как мобильные устройства (Android), веб и настольные компьютеры (MacOS, Linux, Windows). | **Отчетность и логирование:** Playwright не предоставляет гибкие возможности для создания отчетов и логирования результатов тестов.Для этого требуются дополнительные плагины. |
| **Эмуляция мобильных устройств**: Playwright может эмулировать мобильные устройства, включая геолокацию, размер экрана и другие характеристики устройств. | **Совместимость с инфраструктурой**: Playwright плохо интегрируется с CI/CD-пайплайнами, Docker и другими инструментами. |
| **Поддержка нескольких языков программирования**: Изначально созданный для Node.js, Playwright теперь предлагает привязки для JavaScript, TypeScript, Python, Java и C#/.NET, что делает его доступным для широкого круга разработчиков и тестировщиков. |  |
| **Режимы без графического интерфейса и с графическим интерфейсом**: Он может запускать браузеры в режиме без графического интерфейса (headless) для более быстрого выполнения в тестовых средах и в режиме с графическим интерфейсом для разработки и отладки. |  |

Таблица 4. Преимущества и недостатки Playwright