

РОЧЕВ К. В., ЖИФАРСКИЙ В. Д.
АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ ГРАФИЧЕСКИХ ПРИЛОЖЕНИЙ
НА ОСНОВЕ ЗАПИСИ МАКРОСОВ И ВИЗУАЛЬНОГО СРАВНЕНИЯ
ИНТЕРФЕЙСОВ

УДК 004.42:004.05, ГРНТИ 50.05.13

Статья поступила в редакцию 20.06.2025

Автоматизация тестирования
графических приложений на основе
записи макросов и визуального сравнения
интерфейсов

Automation of testing of graphical
applications based on macro
recording and visual comparison of
interfaces

К. В. Рочев, В. Д. Жифарский

K. V. Rochev, V. D. Zhifarsky

Ухтинский государственный технический
университет, г. Ухта

Ukhta State Technical University,
Ukhta

В данной статье рассматривается способ автоматизации ручного тестирования графических приложений посредством записи и автоматического воспроизведения действий пользователя с последующей валидацией результатов тестирования при помощи визуального сравнения состояний интерфейса. Предложенный подход не требует от тестирующего навыков программирования и применим как к веб-приложениям, так и к нативным оконным и игровым приложениям. Описаны архитектура системы, применённые технологии и реализованные функции, а также преимущества предложенного решения по сравнению с существующими инструментами автоматизации тестирования.

This article discusses a method for automating manual testing of graphical applications by recording and automatically reproducing user actions, followed by validation of test results using visual comparison of interface states. The proposed approach does not require programming skills from the tester and is applicable to web applications as well as to native windowed and game applications. The architecture of the system, the technologies employed and the implemented functions are described, together with the advantages of the proposed solution over existing test automation tools.

Ключевые слова: тестирование, автоматизация тестирования, графические приложения, запись макросов, визуальный анализ, сравнение интерфейсов

Keywords: testing, test automation, graphical applications, macro recording, visual analysis, interface comparison

Введение

В условиях постоянного роста требований к качеству программного обеспечения особую роль играет процесс тестирования, позволяющий выявлять ошибки и дефекты на ранних этапах жизненного цикла разработки. Особенно остро эта необходимость проявляется в сфере разработки интерактивных приложений и компьютерных игр, где сложность интерфейсов и большое количество сценариев пользовательского взаимодействия требуют тщательной проверки.

Традиционно в небольших и средних компаниях широко применяется ручное тестирование, при котором тестировщики выполняют последовательности действий в программной среде, фиксируя возникающие ошибки. Однако этот подход имеет ряд существенных недостатков: высокая трудоёмкость, влияние человеческого фактора, невозможность точного воспроизведения тестовых сценариев и ограниченные возможности масштабирования.

Одним из перспективных направлений решения этих проблем является автоматизация тестирования – в частности, запись действий пользователя с последующим воспроизведением макросов. При сочетании этого подхода с визуальным анализом интерфейса возможно существенно повысить эффективность и надёжность проверки программного продукта без необходимости в написании сложных тестовых скриптов.

Настоящая статья посвящена разработке и описанию информационной системы, предназначенной для автоматизации ручного тестирования программного обеспечения на основе перехвата и воспроизведения событий ввода и визуального сравнения состояний интерфейса.

Предпроектное исследование

Целью исследования является создание универсального и гибкого инструмента автоматизированного тестирования, ориентированного на приложения с графическим интерфейсом, в том числе игровые и оконные решения. В статье рассмотрены архитектура системы, применённые технологии, реализованные функции, а также преимущества предложенного подхода по сравнению с существующими решениями.

Современный рынок предлагает большое количество инструментов для автоматизации тестирования программного обеспечения. Наиболее известными являются такие решения, как Selenium, TestComplete, Ranorex, Katalon Studio и др. Каждый из этих инструментов обладает собственным набором возможностей, преимуществ и ограничений, однако при выборе решения для конкретных задач критически важна его применимость к типу тестируемых приложений.

Selenium – одна из самых популярных систем автоматизированного тестирования, ориентированная в первую очередь на веб-приложения. Она обеспечивает поддержку различных браузеров и языков программирования, но не подходит для тестирования нативных оконных или игровых приложений.

Кроме того, для написания тестов требуется опыт программирования и работа с DOM-моделью.

TestComplete – мощная коммерческая система с поддержкой UI-тестирования для различных платформ. Предоставляет удобный интерфейс для записи сценариев, проверки интерфейсов и анализа логов. Однако стоимость лицензии делает её малодоступной для небольших студий и фриланс-разработчиков. Кроме того, её адаптация к нестандартным игровым интерфейсам может требовать ручной доработки.

Ranorex и Katalon Studio – решения, ориентированные на автоматизацию тестирования бизнес-приложений. Они предоставляют поддержку кросс-платформенных тестов и визуальных сценариев, однако в большинстве случаев не справляются с тестированием интерактивных приложений, таких как игры, или со специфичными нативными окнами.

Таким образом, анализ существующих решений показывает, что на рынке отсутствует универсальный инструмент, ориентированный на автоматизацию тестирования нативных или игровых приложений. Это и предопределяет необходимость разработки специализированной информационной системы, адаптированной под задачи компании GlintGate Gaming L.L.C. и не требующей от тестировщиков навыков программирования.

Основной метод, положенный в основу предлагаемой информационной системы, – это автоматизация ручного тестирования путём записи действий пользователя в виде макросов и их последующего автоматического воспроизведения с применением визуального анализа интерфейса.

Суть подхода заключается в следующем:

- система перехватывает действия пользователя (движения мыши, клики, ввод с клавиатуры) с помощью системных хуков Windows API [4];
- эти действия сопровождаются временными метками и снимками экрана;
- полученные данные сохраняются в виде макроса и могут быть использованы для повторного воспроизведения;
- при воспроизведении система эмулирует пользовательский ввод и производит сравнение текущих изображений интерфейса с ранее сохранёнными с помощью библиотеки OpenCV [7];
- по результатам воспроизведения формируется отчёт о прохождении теста, содержащий степень совпадения скриншотов и другие диагностические сведения.

Такой подход позволяет обеспечить:

- точность воспроизведения пользовательского поведения;
- автоматическую валидацию результата без участия тестировщика;
- универсальность: поддержка как веб-приложений, так и десктопных и мобильных приложений, в том числе игр;
- минимизацию требований к квалификации персонала – интерфейс не требует написания кода

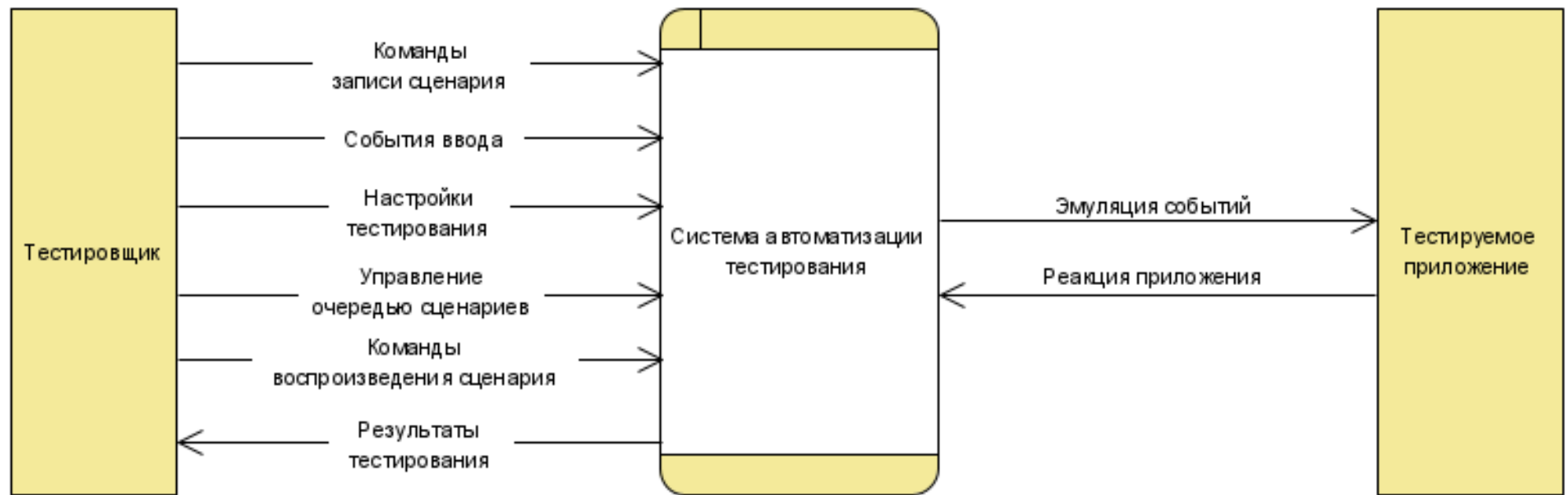


Рисунок 1. Контекстная диаграмма «как будет»

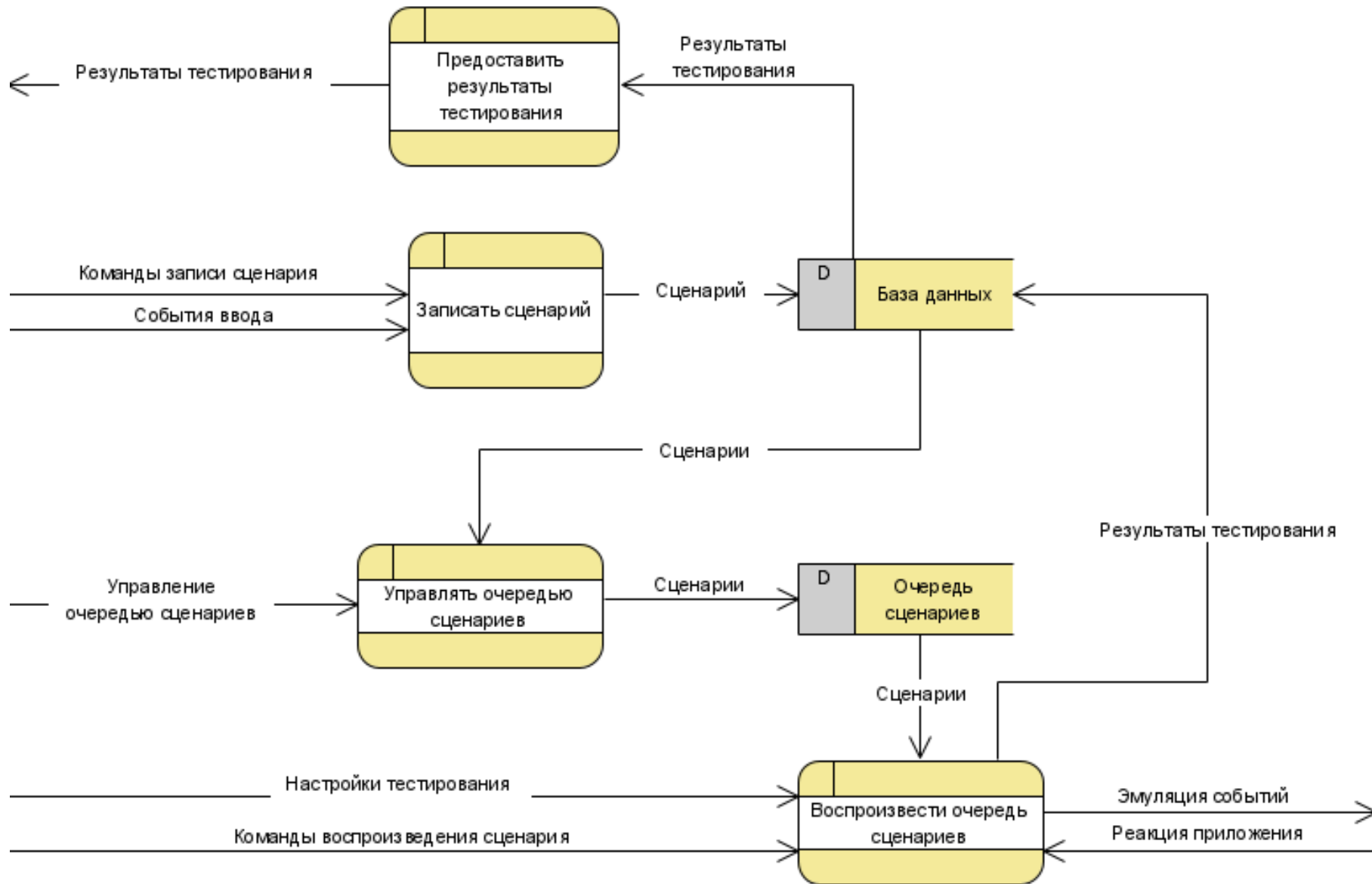


Рисунок 2. Диаграмма потоков данных «как будет» 1-го уровня

Проектирование системы

При проектировании системы были выделены следующие сущности:

– пользователь (тестировщик) – создает сценарии тестирования, инициирует запуск тестирования, получает и анализирует результаты тестирования;

тестируемое приложение – тестируемое программное обеспечение. Получает последовательность действий в соответствии со сценарием тестирования. Реакция приложения фиксируется в системе

При декомпозиции были выделены следующие процессы:

- записать сценарий;
- управлять очередью сценариев;
- воспроизвести очередь сценариев;
- предоставить результаты тестирования.

Разрабатываемая система построена на модульной архитектуре, обеспечивающей надёжность, масштабируемость и лёгкость сопровождения.

Основные компоненты системы:

1. Ядро системы – управляет логикой записи, воспроизведения макросов, обработкой событий ввода и взаимодействием с другими модулями.

2. Графический интерфейс – реализован на ImGui [5], обеспечивает доступ к функциям записи, загрузки, воспроизведения сценариев, а также отображение отчётов.

3. Внедряемая библиотека (DLL)

- ✓ Подгружается в процесс тестируемого приложения;
- ✓ Перехватывает действия пользователя через системные хуки;
- ✓ Отправляет события в ядро системы через именованные каналы.

4. Слой доступа к данным

- ✓ Сохраняет макросы и результаты тестирования в базу данных;
- ✓ Реализован в виде абстрактного интерфейса, поддерживающего различные СУБД.

5. Модуль визуального анализа – выполняет сравнение изображений при помощи библиотеки OpenCV.

Реализация

Реализация системы выполнена на языке C++, с использованием Microsoft Visual Studio. В качестве графического интерфейса применена библиотека ImGui, что позволило быстро реализовать удобный пользовательский интерфейс. Для визуального анализа и сравнения изображений использована библиотека OpenCV, а для взаимодействия между модулями – Named Pipes (именованные каналы) [6].

Информационная система автоматизации тестирования была реализована как настольное приложение, ориентированное на тестировщиков, не обладающих навыками программирования. Основное внимание уделено

простоте интерфейса, удобству запуска тестов и точности воспроизведения сценариев.

Запись сценария начинается с запуска исполняемого файла тестируемого приложения (Рисунок 3).

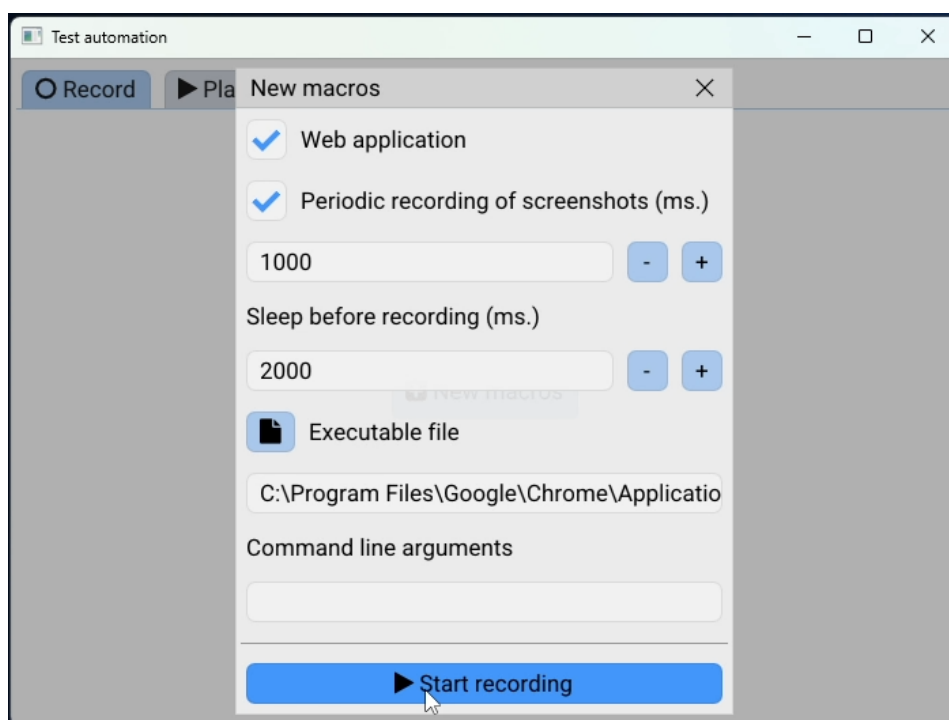


Рисунок 3. Окно «Новый сценарий»

Система:

- встраивает библиотеку (DLL) в процесс приложения для перехвата ввода;
- устанавливает хуки на клавиатуру и мышь для перехвата событий;
- фиксирует координаты и тип события, временные метки, снимки интерфейса приложения;
- формирует макрос, содержащий полученные при записи данные;
- позволяет просмотреть результаты записи и сохранить сценарий (Рисунок 4).

В режиме воспроизведения (Рисунок 5):

- действия из макроса эмулируются с помощью Windows API (функция `SendInput`);
- окну приложения автоматически задаются размеры и положение, соответствующие моменту записи;
- система сохраняет снимки интерфейса приложения и сравнивает их с эталонными;
- при отклонении отображается информация о несовпадении.

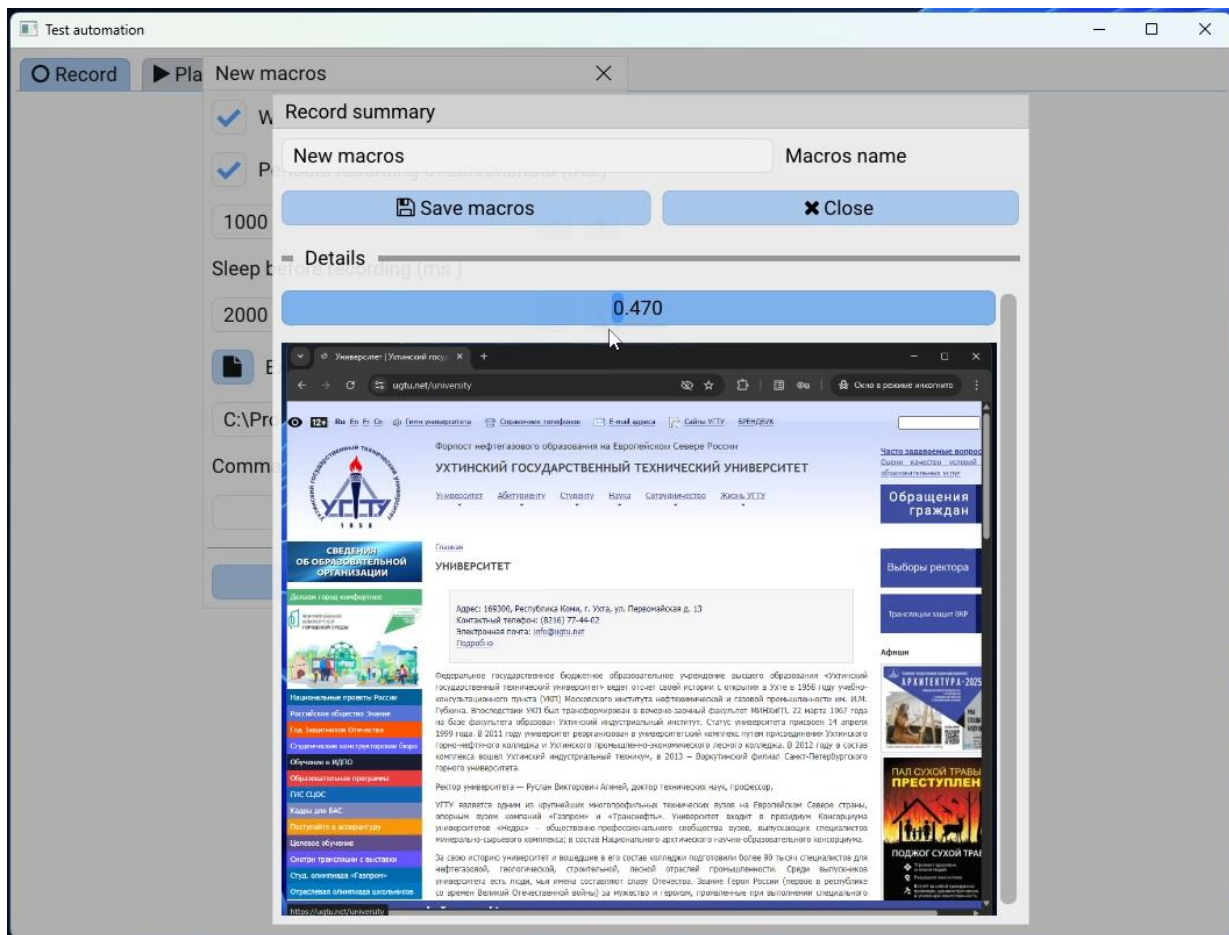


Рисунок 4. Окно с результатами записи

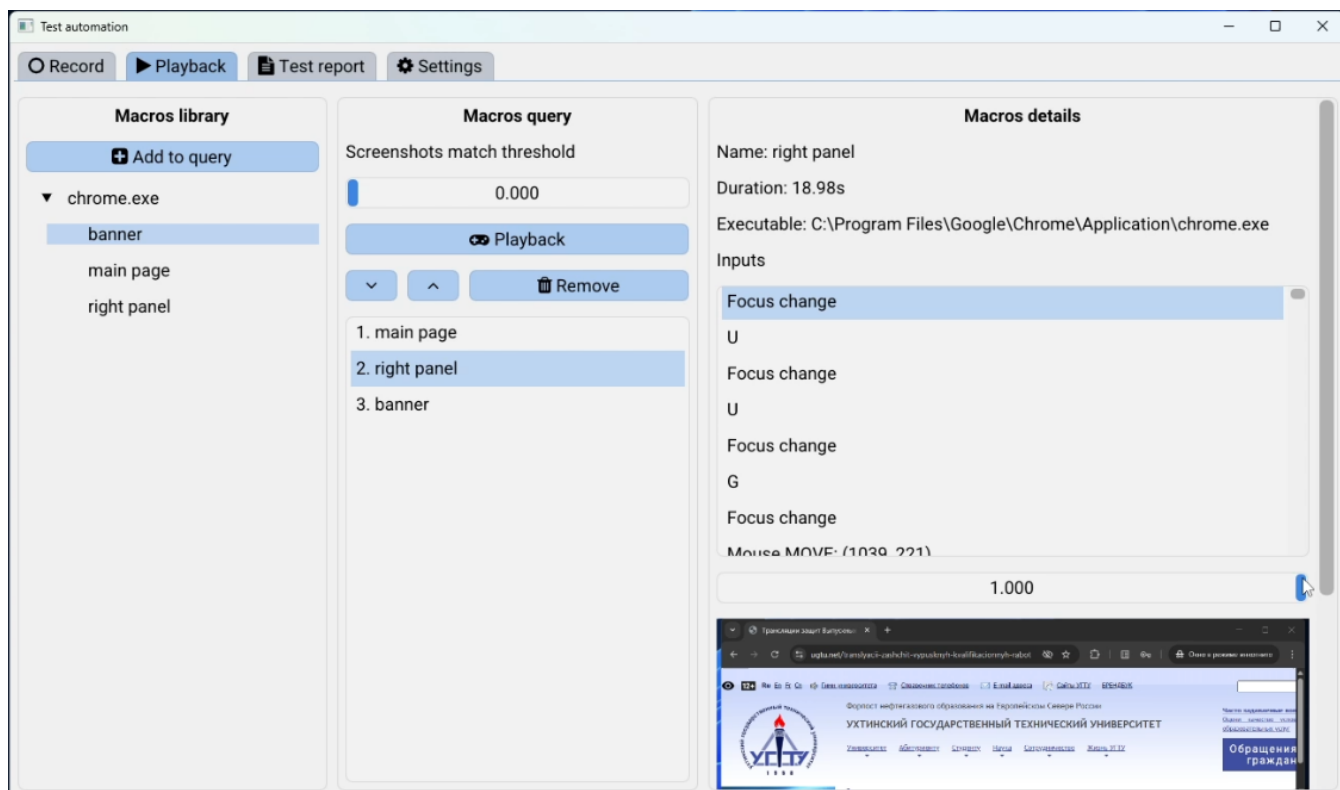


Рисунок 5. Вкладка «Воспроизведение»

Для тестирования нескольких сценариев подряд реализована очередь сценариев.

Пользователь может:

- добавлять/удалять сценарии;
- менять порядок воспроизведения;
- запускать тестирование всей очереди одной кнопкой;
- по завершении получить единый отчёт, включающий результаты всех тестов.

По завершении тестирования система формирует наглядный отчёт (Рисунок 6), содержащий:

- список выполненных сценариев;
- длительность каждого теста;
- процент совпадения скриншотов;
- метки об ошибках или отклонениях;
- возможность открыть подробности и сравнить изображения вручную.

Таким образом, даже при автоматическом воспроизведении сохраняется возможность ручной верификации и принятия решения тестирующим.

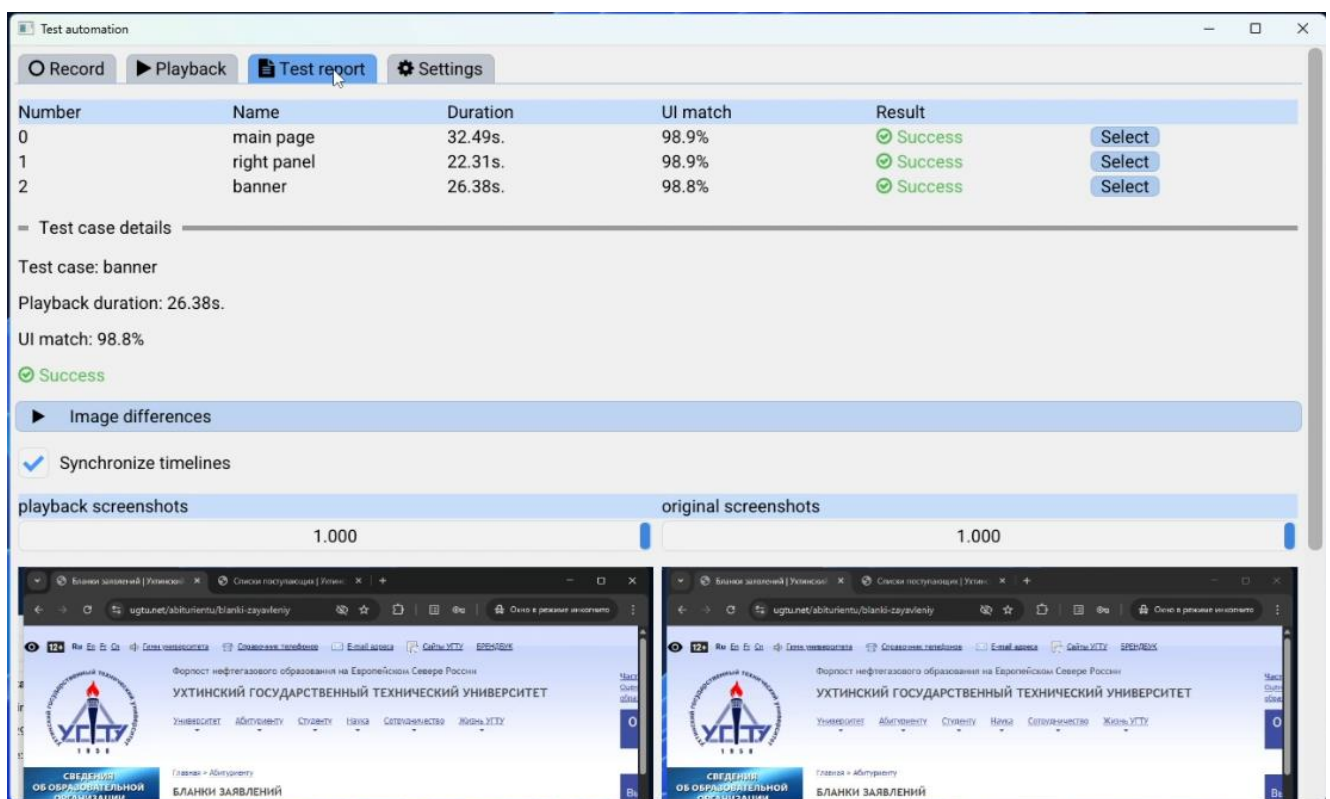


Рисунок 6. Отчет о тестировании

Заключение

Система продемонстрировала стабильную работу при высоких нагрузках, а её архитектура позволяет масштабировать решение и адаптировать под другие проекты компании без существенных затрат.

Предложенное решение позволяет:

- автоматизировать ручные сценарии тестирования без программирования;
- обеспечить точное воспроизведение пользовательских действий;
- использовать визуальное сравнение интерфейсных состояний для оценки корректности работы;
- существенно сократить временные и трудовые затраты на тестирование.

Разработанная архитектура отличается модульностью и открытостью для расширения. В перспективе возможно дополнение системы средствами анализа логов, интеграцией с CI/CD-пайплайнами, а также добавлением инструментов на базе искусственного интеллекта для предиктивной аналитики и автоматического формирования сценариев.

Таким образом, созданная система представляет собой универсальный инструмент для автоматизации тестирования, ориентированный на гибкость, точность и удобство, и может быть полезна как для внутренних нужд IT-компаний, так и в образовательных целях.

Список использованных источников и литературы

1. Тюшнякова И. А., Пискунова Н. Автоматизация тестирования программного обеспечения: ключевые методологии и подходы // Вестник Таганрогского института имени А. П. Чехова. 2024. №2.
2. Артюхова А. С. Проблемы автоматизации тестирования и подходы к их решению // CETERIS PARIBUS. 2016. №10.
3. Макаров К. С., Фаткин Р. И. Скриншотное тестирование как многоаспектный вид автоматизированной динамической верификации вебприложений // Программные системы и вычислительные методы. 2025. №1.
4. Рочев К. В. Классификация средств графического моделирования для разработки информационных систем // Информационные технологии в управлении и экономике. 2024. №1. Режим доступа: <http://itue.ru/Issue/Article/275>
5. Документация Microsoft [Электронный ресурс]. – URL: <https://learn.microsoft.com> (дата обращения: 23.06.2025).
6. ImGui Documentation [Электронный ресурс]. – URL: <https://github.com/ocornut/imgui> (дата обращения: 23.06.2025).
7. MSDN Library – Named Pipes [Электронный ресурс]. – URL: <https://learn.microsoft.com/en-us/windows/win32/ipc/named-pipes> (дата обращения: 23.06.2025).
8. Официальная документация OpenCV [Электронный ресурс]. – Режим доступа: <https://docs.opencv.org/4.x/index.html> (дата обращения: 23.06.2025).

List of references

1. Tyushnyakova I. A., Piskunova N. Automation of Software Testing: Key Methodologies and Approaches // Bulletin of the Taganrog Institute named after A.P. Chekhov. 2024. No. 2.
2. Artyukhova A. S. Problems of Test Automation and Approaches to Their Solution // CETERIS PARIBUS. 2016. No. 10.
3. Makarov K. S., Fatkin R. I. Screenshot Testing as a Multifaceted Type of Automated Dynamic Verification of Web Applications // Software Systems and Computational Methods. 2025. No. 1.
4. Rochev K. V. Classification of Graphical Modeling Tools for Information Systems Development // Information Technologies in Management and Economics. 2024. No. 1. Available at: <http://itue.ru/Issue/Article/275>
5. Microsoft Documentation [Online]. – URL: <https://learn.microsoft.com> (accessed: 23.06.2025).
6. ImGui Documentation [Online]. – URL: <https://github.com/ocornut/imgui> (accessed: 23.06.2025).
7. MSDN Library – Named Pipes [Online]. – URL: <https://learn.microsoft.com/en-us/windows/win32/ipc/named-pipes> (accessed: 23.06.2025).
8. OpenCV Official Documentation [Online]. – Available at: <https://docs.opencv.org/4.x/index.html> (accessed: 23.06.2025).