



АВТОМАТИЗАЦИЯ ТЕСТИРОВАНИЯ С ИСПОЛЬЗОВАНИЕМ ИНСТРУМЕНТОВ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

*Хусаинова Эльвира Робертовна,
инженер по автоматизации тестирования,
Deutsche Telekom ITTS, Венгрия, г. Будапешт*

E-mail: elechka.de@gmail.com

Аннотация. В статье рассматриваются возможности и перспективы применения инструментов искусственного интеллекта (ИИ) в автоматизации тестирования программного обеспечения. Проведен комплексный анализ архитектурных, методологических и практических аспектов внедрения ИИ в процессы обеспечения качества, включая CI/CD-конвейеры. Отмечены ключевые подходы, применяемые на различных уровнях тестирования. Рассматриваются проблемы интерпретируемости ИИ-решений, подготовки обучающих выборок, а также вопросы этической обработки пользовательских данных. Делается вывод о необходимости комплексного внедрения DataOps и XAI-подходов для устойчивого развития интеллектуального тестирования.

Ключевые слова: искусственный интеллект, автоматизация тестирования, синтетические данные, оркестрация тестов, предсказание дефектов.

Актуальность исследования

В условиях стремительного развития информационных технологий и увеличения сложности программных продуктов, обеспечение высокого качества программного обеспечения становится приоритетной задачей для IT-компаний. Традиционные методы тестирования, особенно ручные, не всегда способны эффективно справляться с объемами и скоростью изменений в современных системах.

Внедрение искусственного интеллекта (ИИ) в процессы автоматизации тестирования открывает новые возможности для повышения эффективности и точности тестирования. ИИ способен анализировать большие объемы данных, предсказывать потенциальные дефекты и автоматически адаптировать тестовые сценарии к изменениям в программном обеспечении. Такие возможности особенно актуальны в условиях Agile и DevOps-подходов, где требуется быстрая и непрерывная интеграция и доставка программных продуктов.

Таким образом, изучение и внедрение ИИ в автоматизацию тестирования является актуальным направлением, способным повысить качество и скорость разработки программных продуктов.

Цель исследования

Целью данного исследования является анализ возможностей и перспектив использования инструментов искусственного интеллекта в автоматизации тестирования программного обеспечения.

Материалы и методы исследования

В качестве основных материалов исследования использованы научные публикации по теме интеллектуального тестирования, обзоры инструментов автоматизации, отчеты отраслевых лидеров.

Методологическая база включает: аналитико-сравнительный метод, системный подход, контент-анализ.

Результаты исследования

Одним из ключевых вызовов современного программного обеспечения является обеспечение его надежности и качества при постоянно возрастающей сложности и ускоряющихся релизных циклах.

Алгоритмы искусственного интеллекта могут быть чрезвычайно полезны в индустрии тестирования для создания более умного и более производительного программного обеспечения для конечного пользователя. Тем не менее, важно понять, как правильно использовать ИИ. Алгоритмы, которые работают как настоящий пользователь, обращающийся к автоматизации. После этого необходимо определить области внутри процесса, которые можно оптимизировать с помощью ИИ, и включить алгоритм. Наличие интеллектуального алгоритма может облегчить этот процесс и помочь тестировщикам найти максимальное количество ошибок [4, с. 187].

В условиях внедрения методологий DevOps и Agile возрастает потребность в непрерывной интеграции и доставке программного продукта (CI/CD), что требует автоматизации процессов тестирования. Однако традиционные средства автоматизации, такие как Selenium, Appium и TestNG, хотя и повышают производительность тестирования, зачастую оказываются недостаточными для адаптивного и интеллектуального анализа поведения системы. Именно в этом контексте актуализируется применение инструментов искусственного интеллекта (ИИ) в автоматизации тестирования.

Определение понятия искусственного интеллекта во многих источниках сводится к тому, что искусственный интеллект (ИИ) – это способность технического устройства или робота, управляемого компьютерной системой, выполнять задачи, подлежащие решению рациональным человеческим интеллектом. Такие устройства или системы должны обладать свойствами, характерными для человеческого разума [1, с. 50].

Уже в 2017 г., благодаря опросам, проведенным Dell Technologies и institute for the Future, выяснилось, что примерно 1% всех компаний использует у себя искусственный интеллект. После проведения анкетирования директоров крупных компаний было установлено, что увеличилось число применяемых преимуществ искусственного интеллекта, а именно – почти 37% предприятий задействуют его преимущества. Вполне логично видеть показатель в 90%

респондентов, которые заявляют об использовании искусственного интеллекта у себя к 2022 г., согласно опросу Cognilytica [2, с. 60].

ИИ в области тестирования ПО включает в себя использование алгоритмов машинного обучения (ML), обработки естественного языка (NLP), логического вывода и анализа поведения пользователей для создания, адаптации и оптимизации тестовых сценариев. В отличие от традиционной автоматизации, ИИ способен выявлять закономерности в коде, анализировать логи, предсказывать уязвимости и самостоятельно формировать новые тест-кейсы.

Наиболее распространенные функции ИИ в тестировании включают автоматическое распознавание изменений UI (self-healing tests), предсказание дефектов, генерацию тестов на основе пользовательских сценариев и приоритизацию багов по степени влияния на пользователя.

Показательным примером успешной интеграции ИИ в QA-процесс является кейс компании Facebook, где применяется собственная система Sapienz. Она использует эволюционные алгоритмы для автоматического тестирования мобильных приложений и выявления сбоев при взаимодействии между компонентами. Согласно внутреннему отчету, применение Sapienz позволило сократить на 37% количество инцидентов, доходящих до продакшн-среды.

Практические преимущества ИИ включают:

1. Скорость. В среднем время прохождения полного тест-ранна сокращается на 40-60%;
2. Адаптивность. Self-healing тесты автоматически адаптируются к изменившейся верстке;
3. Покрытие. AI-инструменты способны генерировать сценарии, охватывающие до 90% пользовательских маршрутов.

Искусственный интеллект – это не система, но он реализован в системе. Вообще говоря, это обобщающий термин, используемый для отрасли компьютерных наук. Он означает способность приобретать и применять знания, а также действовать естественно, как человек. Например, планирование, решение проблем и рассуждение [5, с. 29].

Однако существует ряд вызовов. Во-первых, интерпретируемость решений ИИ по-прежнему ограничена: не всегда можно точно объяснить, почему та или иная аномалия распознана как дефект. Во-вторых, качество данных, используемых для обучения моделей, напрямую влияет на точность предсказаний. Также, высокая стоимость внедрения и необходимость подготовки специалистов с гибридной компетенцией (AI + QA) тормозят широкомасштабное применение в компаниях малого и среднего бизнеса.

В таблице 1 представлены сравнительные характеристики традиционной автоматизации и ИИ-автоматизации.

технической инфраструктуры, но и подходов к построению процессов тестирования, включая DataOps, ML Ops и TestOps парадигмы.

Одним из ключевых элементов является предобработка и аннотирование данных для обучения моделей. В отличие от традиционной автоматизации, где

тест-кейсы задаются вручную, ИИ требует больших объемов исторических данных: логов выполнения, журналов ошибок, данных пользовательского взаимодействия. Внедрение инфраструктуры сбора и стандартизации этих данных – фундаментальный шаг для создания обучающего датасета. В организациях, работающих по принципу Data-Driven QA, такие данные используются для построения моделей предсказания дефектов и генерации сценариев наиболее критичных путей прохождения по приложению.

Рисунок 1 отражает динамику роста внедрения ИИ в автоматизацию тестирования на протяжении последних лет (по данным аналитики Deloitte и DevOps Institute).

Таблица 1

**Сравнительные характеристики
традиционной автоматизации и ИИ-автоматизации**

Критерий	Традиционная автоматизация	Автоматизация с использованием ИИ
Создание тестов	Требует ручного написания скриптов с использованием языков программирования и фреймворков (например, Selenium, JUnit)	Автоматическая генерация тест-кейсов на основе анализа требований, пользовательских историй и исторических данных
Обслуживание тестов	Высокие затраты на поддержку: любые изменения в интерфейсе или логике приложения требуют ручного обновления тестов	Самовосстанавливающиеся тесты: ИИ автоматически адаптирует тесты к изменениям в приложении, снижая необходимость в ручном вмешательстве
Гибкость и адаптивность	Ограниченная гибкость: тесты жестко привязаны к текущей реализации и плохо адаптируются к изменениям	Высокая адаптивность: ИИ способен обучаться на основе новых данных и адаптировать тесты к изменениям в приложении
Обнаружение дефектов	Основано на заранее определенных сценариях; может не выявлять неожиданные ошибки или аномалии	Использует анализ данных и машинное обучение для предсказания и обнаружения потенциальных дефектов, включая аномалии, которые трудно выявить традиционными методами
Требования к ресурсам	Требует усилий QA-инженеров для написания и поддержки тестов	Снижает нагрузку на QA-инженеров за счет автоматизации рутинных задач, позволяя сосредоточиться на более сложных аспектах тестирования
Скорость выполнения	Зависит от объема тестов и может быть ограничена из-за необходимости ручного вмешательства при изменениях	Более высокая скорость за счет параллельного выполнения тестов и автоматической адаптации к изменениям
Затраты на внедрение	Ниже начальные затраты, но выше операционные из-за постоянной необходимости в обслуживании	Более высокие начальные инвестиции в инструменты и обучение, но потенциально ниже операционные затраты благодаря автоматизации

Кроме архитектурных аспектов, в фокусе внимания исследователей находятся и метрики оценки эффективности ИИ в тестировании. Стандартные показатели дополняются новыми метриками:

1. Defect Detection Rate (DDR) – метрика, которая показывает, насколько эффективно искусственный интеллект выявляет ошибки или проблемы в программном обеспечении во время автоматизированного тестирования.

2. Test Scenario Diversity Score (TSDS) – степень уникальности сгенерированных тестов.

3. Self-healing Effectiveness Ratio (SHER) – коэффициент успешного автоматического восстановления тестов после изменений в коде.



Рис. 1 Рост применения ИИ в автоматизации тестирования

Важной тенденцией является развитие Explainable AI (XAI) в тестировании, позволяющего сделать работу ИИ-моделей интерпретируемой для QA-инженеров. Это особенно актуально для регламентированных отраслей – например, финансовой и медицинской, где обоснование результата критично. Существует практика использования таких фреймворков, как SHAP и LIME, позволяющих визуализировать вклад каждого признака в принятие решения о наличии дефекта.

Отдельно стоит отметить практику объединения ИИ с RPA (Robotic Process Automation), что позволяет автоматизировать не только тестирование, но и процессы управления дефектами, логирования и составления отчётности. Примером может служить связка UiPath с AI Builder от Microsoft, применяемая в крупных корпоративных системах для автоматизации end-to-end процессов тестирования.

Кроме того, в научной литературе всё чаще рассматриваются вопросы этического применения ИИ в тестировании, в частности, при анализе пользовательских данных и логов. Важно соблюдение норм конфиденциальности, особенно при использовании данных реальных пользователей для обучения моделей.

Подход к внедрению ИИ-инструментов на каждом из уровней отличается как по архитектуре, так и по целям.

ИИ в тестировании программного обеспечения использует следующие ключевые типы моделей:

1. Supervised learning – обучение с учителем применяется в задачах предсказания дефектов. Например, модели классификации (Logistic Regression, Random Forest, XGBoost) обучаются на метках багов в истории проекта и могут оценивать вероятность дефекта для новых компонентов.

2. Unsupervised learning – используется для кластеризации сценариев использования и выявления аномального поведения пользователя. Такие модели применяются в динамическом тестировании, например, для автоматического выделения edge-case сценариев.

3. Reinforcement learning – перспективный подход, при котором агент обучается генерировать эффективные тест-кейсы, максимизируя покрытие или количество найденных багов. Эта техника применяется в исследовательских проектах, таких как DeepTest от IBM Research.

4. Natural Language Processing (NLP) – используется в генерации тест-кейсов из требований, описанных на естественном языке. Например, система может распознать фразу «пользователь должен получить письмо с подтверждением» и автоматически построить проверку этого поведения.

Таблица 2 систематизирует, как и для чего используется ИИ на каждом уровне.

Таблица 2

ИИ на разных уровнях тестирования

Уровень тестирования	Роль ИИ	Примеры применения
Unit tests	Выявление функций без тестов, генерация на основе AST	Diffblue Cover (анализ Java-кода)
Integration tests	Анализ API-взаимодействий, генерация контрактов	RESTler от Microsoft (автоматическое API fuzzing)
System tests	Генерация пользовательских сценариев на основе логов	Functionize, Mabl
Acceptance tests	Преобразование требований в тест-кейсы с использованием NLP	Gherkin-based генераторы

Современные средства в сфере информационных технологий позволяют активизировать и развить точность использования интеллектуальной составляющей профессиональных алгоритмов [3, с. 280].

Тестовые оркестрации с участием ИИ перестают быть «инструментами поддержки» и превращаются в полноценные модули в CI/CD-цепочках. Типовая архитектура включает следующие уровни:

- слой наблюдаемости – сбор логов, трассировок, метрик;
- слой хранилищ – централизованное хранилище исторических данных о тестировании;

- слой обработки данных – предобработка и фиче-инжиниринг для обучения моделей;
- Inference Layer – применение уже обученных моделей в реальном времени (например, при каждом новом коммите в Git);
- Orchestration Engine – принятие решений о запуске и приоритизации тестов.

В современных компаниях практикуется использование Kubernetes-платформ для разворачивания микросервисной архитектуры тестирования с отдельными сервисами:

- API-сервис предсказаний (в контейнере с моделью);
- сервис очередей;
- сервис дашбордов;
- MLflow-сервер для управления экспериментами.

Одной из ключевых проблем внедрения ИИ в CI/CD является подготовка валидных и этически допустимых наборов данных для обучения моделей. Здесь вступает в силу методология DataOps, заимствованная из практик Data Science. Она предполагает:

- версионирование тестовых данных (через DVC, DeltaLake);
- деперсонализацию и анонимизацию логов пользователей (важно для GDPR);
- генерацию synthetic data для редких сценариев (например, edge-case UI-потоков).

Synthetic Data – это искусственно созданные данные, которые имитируют наблюдения реального мира. Они используются для подготовки моделей машинного обучения, когда получение реальных данных невозможно из-за сложности или дороговизны.

Компании, такие как Tonic.ai и Mostly AI, специализируются на генерации синтетических данных, которые близки к реальным по структуре и статистическим свойствам, но не содержат конфиденциальной личной информации.

Некоторые преимущества использования синтетических данных в тестировании:

1. Устранение рисков, связанных с конфиденциальностью. Использование реальных данных для тестирования может привести к раскрытию конфиденциальной информации пользователей.

2. Возможность моделирования редких, но важных событий. Синтетические данные позволяют имитировать ситуации, которые могут не присутствовать в доступных реальных данных.

3. Масштабируемость. Синтетические данные позволяют создавать разнообразные сценарии, которые могут возникнуть в будущем, что позволяет приложению адаптироваться к новым требованиям.

Выводы

Автоматизация тестирования с использованием инструментов искусственного интеллекта представляет собой перспективное направление развития программной инженерии, позволяющее повысить качество, адаптивность и эффективность процессов обеспечения качества ПО. Внедрение ИИ в тестирование обеспечивает переход от статических, скриптовых методов к динамическим, самообучающимся системам, способным автоматически адаптировать тестовые сценарии к изменениям в коде, предсказывать дефекты, выявлять аномалии и повышать полноту тестового покрытия.

Несмотря на высокие технологические возможности, повсеместное внедрение ИИ сталкивается с рядом ограничений: от высокой стоимости внедрения и нехватки квалифицированных специалистов до проблем интерпретируемости моделей и необходимости соблюдения стандартов конфиденциальности. Эти вызовы требуют внедрения explainable AI-подходов, прозрачных стратегий управления данными и адаптации процессов тестирования к новым парадигмам DevTestOps.

Таким образом, ИИ-технологии в сфере автоматизации тестирования не только расширяют технический инструментарий QA-инженеров, но и трансформируют саму концепцию обеспечения качества, приближая её к когнитивным, самоадаптирующимся системам, способным функционировать в условиях высокой сложности и непрерывных изменений.

Литература:

1. Ефимова С.А. Развитие искусственного интеллекта // Цифровая наука. – 2020. – № 6. – С. 49-58.
2. Журавлев И.А., Кузнецов А.А. // Переход на искусственный интеллект и его влияние на сотрудников компании / Наука и бизнес: пути развития. – 2021. – № 1(115). – С. 60-63.
3. Масюк Н.Н., Васюкова Л.К., Бушуева М.А., Диденко П.С. Нейронные сети как прорывная цифровая технология в инновационном бизнесе // Азимут научных исследований: экономика и управление. – 2019. – Т. 8, № 4(29). – С. 279-283. – DOI 10.26140/anie-2019-0804-0062.
4. Тулфоров Д.М. Внедрение искусственного интеллекта в тестирование программного обеспечения // Наука третьего тысячелетия: материалы Международной (заочной) научно-практической конференции. – 2020. – С. 184-189.
5. Чомаев Ш.У.А. Искусственный интеллект: понятие, задачи области применения // Актуальные исследования. – 2021. – № 30(57). – С. 28-30.