

ОБЗОР И СРАВНИТЕЛЬНЫЙ АНАЛИЗ МЕТОДОВ АВТОМАТИЗАЦИИ ТЕСТИРОВАНИЯ UX ДЛЯ БИЗНЕС-ПРИЛОЖЕНИЙ

Гугалев А.С.

*Белорусский государственный университет информатики и радиоэлектроники,
г. Минск, Республика Беларусь*

Научный руководитель: Косарева Е.М. – ассистент кафедры ПИКС

Аннотация. В статье проводится сравнительный анализ трёх методов автоматизации контроля пользовательского опыта (UX) в корпоративных информационных системах: инструментального аудита доступности, визуального регрессионного тестирования и мониторинга воспринимаемой производительности. Для каждого метода определены измеримые критерии оценки – автоматизируемость, стабильность тестов и влияние выявляемых дефектов на работу конечного пользователя. На основе анализа данных отраслевых стандартов W3C и документации ведущих инструментов автоматизации обоснована рациональная стратегия обеспечения качества UX для бизнес-ориентированных программных продуктов.

Ключевые слова: автоматизация UX, бизнес-приложения, UI-тестирование, визуальное тестирование, WCAG, Lighthouse, пользовательский опыт

Введение. В сегменте корпоративного программного обеспечения (ERP, CRM, банковские системы) качество пользовательского интерфейса напрямую влияет на операционную эффективность организации. Сложные многоступенчатые формы и перегруженные данными экраны повышают когнитивную нагрузку на персонал и увеличивают вероятность ошибок ввода [1]. В условиях непрерывной поставки программного обеспечения (CI/CD) традиционные методы ручного UX-тестирования становятся узким местом: их трудоёмкость несовместима с темпом коротких итераций [2].

Ключевая проблема автоматизации UX-контроля состоит в операционализации субъективного понятия «удобство» через измеримые технические метрики, допускающие верификацию программными средствами без участия человека на этапе регрессионного тестирования [3]. Целью настоящей работы является систематизация и сравнительная оценка трёх доминирующих методов автоматизации UX-контроля по критериям, значимым для бизнес-ориентированных систем

Основная часть. Переход к автоматизированному контролю UX основан на инженерном подходе к юзабилити, согласно которому интерфейс должен соответствовать измеримым нормативным требованиям по доступности, визуальной целостности и отзывчивости [4].

В современной практике выделяются три самостоятельных метода автоматизации UX-контроля:

1 Инструментальный аудит доступности (Accessibility-as-Code). Метод основан на программной проверке соответствия интерфейса стандарту WCAG 2.1: инструменты типа Axe-core встраиваются в конвейер сборки и автоматически проверяют контрастность, семантическую разметку и структуру DOM. По данным исследования Deque Systems, охватившего более 13 000 страниц реальных приложений, такие средства выявляют в среднем 57% проблем доступности без участия человека [5]; оставшиеся ~43% (сложные сценарии клавиатурной навигации, контекстные ошибки восприятия) требуют ручной экспертизы.

2 Визуальное регрессионное тестирование (Visual Regression Testing). Инструменты попиксельного сравнения (Percy, Appliflow Eyes) фиксируют эталонные снимки интерфейса и при каждом прогоне обнаруживают отклонения: смещения элементов, изменения шрифтов, нарушения вёрстки [6]. Метод незаменим при масштабных обновлениях библиотек UI-компонентов, однако незначительные визуальные изменения, не влияющие на работу

пользователя, способны вызвать ложноположительные срабатывания, устранение которых требует регулярного обновления базовых снимков и повышает стоимость поддержки в Agile-среде [7].

3 Мониторинг метрик воспринимаемой производительности (Perceived Performance Metrics). Поскольку скорость реакции системы напрямую определяет продуктивность персонала, UX неотделим от производительности интерфейса. Google Lighthouse измеряет метрики LCP, TBT и CLS – три из пяти ключевых показателей актуальной методологии Lighthouse 10 [8] – и позволяет своевременно обнаруживать снижение отзывчивости интерфейса по мере усложнения бизнес-логики. Ограничение метода – чувствительность результатов к нагрузке на тестовое окружение, что требует стабилизации инфраструктуры для получения воспроизводимых значений.

Для сравнительного анализа выбраны три критерия, обусловленные спецификой корпоративной эксплуатации: автоматизируемость (доля дефектов, обнаруживаемых без участия человека), стабильность тестов (частота ложноположительных срабатываний и стоимость поддержки при частых релизах) и влияние на UX (критичность дефектов для выполнения бизнес-операций: «критическое» – полная блокировка работы пользователя, «высокое» – существенное затруднение навигации, «среднее» – замедление без блокировки).

По автоматизируемости аудит доступности и мониторинг производительности оцениваются как «высокие»: Axe-core выявляет 57% дефектов автоматически [5], Lighthouse полностью автоматизирует сбор LCP, TBT, CLS [8]; визуальный регрессионный метод – «средний», так как требует ручной разметки эталонов [6]. По стабильности лидирует аудит доступности – проверки детерминированы и опираются на формальные правила WCAG; мониторинг производительности занимает среднюю позицию из-за чувствительности к нагрузке окружения; визуальное тестирование наименее стабильно [7]. По влиянию на UX нарушения доступности критичны – они способны полностью исключить категорию пользователей из работы с системой [3]; визуальные дефекты существенно затрудняют навигацию, но редко блокируют задачу; снижение производительности замедляет бизнес-процессы и повышает вероятность ошибок оператора [1].

Сводные результаты приведены в таблице 1.

Таблица 1 – Сравнительный анализ методов автоматизации UX

Метод	Основная проверяемая метрика	Автоматизируемость	Стабильность тестов	Влияние на UX
Accessibility Audit	WCAG 2.1 compliance score	Высокая [5]	Высокая	Критическое
Visual Regression	Visual Consistency Score	Средняя	Низкая [7]	Высокое
Performance Audit	LCP, TBT, CLS (Lighthouse 10)	Высокая [8]	Средняя	Среднее

Анализ данных таблицы 1 показывает, что аудит доступности обеспечивает наилучшее сочетание высокой автоматизируемости и стабильности при максимальной критичности выявляемых дефектов для конечного пользователя, что делает его приоритетным методом для обязательного включения в CI/CD-конвейер. Мониторинг производительности сопоставим по уровню автоматизации, но меньшая стабильность результатов и менее критичный характер дефектов определяют его роль как дополнительного слоя контроля, целесообразного при усложнении бизнес-логики приложения. Визуальный регрессионный анализ оправдан при масштабных обновлениях дизайн-системы или библиотек UI-компонентов, но высокие затраты на поддержку эталонной базы ограничивают его применение в качестве рутинного инструмента в условиях коротких итераций. В совокупности два высокоавтоматизируемых метода – аудит доступности и мониторинг производительности – способны выявить значительную долю UX-дефектов ещё до начала ручного тестирования [2], позволяя

специалистам по качеству сосредоточиться на проверке сложных пользовательских сценариев, не поддающихся формализации.

Заключение. В результате проведённого исследования установлено, что автоматизация UX-тестирования позволяет перевести субъективные оценки удобства интерфейса в область измеримых технических показателей. Из трёх рассмотренных методов инструментальный аудит доступности демонстрирует наилучшее соотношение автоматизируемости, стабильности и критичности обнаруживаемых дефектов, что обосновывает его включение в CI/CD-конвейер в качестве обязательного шага. Мониторинг метрик воспринимаемой производительности целесообразно применять как дополнительный слой контроля. Визуальное регрессионное тестирование рекомендуется использовать точно – при обновлениях дизайн-системы. Комбинирование первого и третьего методов обеспечивает гарантированный базовый уровень качества интерфейса, минимизирует риски критических ошибок взаимодействия и позволяет специалистам по тестированию сосредоточиться на верификации сложных пользовательских сценариев, не поддающихся формализации.

Список литературы

1. Crispin, L. *Agile Testing: A Practical Guide for Testers and Agile Teams* / L. Crispin, J. Gregory. – Boston : Addison-Wesley, 2009. – 536 p. – ISBN 978-0-321-53446-0.
2. Humble, J. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation* / J. Humble, D. Farley. – Boston : Addison-Wesley, 2010. – 512 p. – ISBN 978-0-321-60191-3.
3. W3C. *Web Content Accessibility Guidelines (WCAG) 2.1* / World Wide Web Consortium // W3C Recommendation [Электронный ресурс]. – Режим доступа: <https://www.w3.org/TR/WCAG21/> (дата обращения: 03.03.2026).
4. Nielsen, J. *Usability Engineering* / J. Nielsen. – Boston : Academic Press, 1993. – 362 p. – ISBN 978-0-12-518406-9.
5. Deque Systems. *Automated Accessibility Coverage Report: How Much Can Automation Catch?* / Deque Systems // Deque Research [Электронный ресурс]. – Режим доступа: <https://www.deque.com/automated-accessibility-coverage-report/> (дата обращения: 03.03.2026).
6. Applitools. *The State of Visual AI Report 2023* / Applitools // Applitools Research [Электронный ресурс]. – Режим доступа: <https://applitools.com/state-of-visual-ai-report/> (дата обращения: 03.03.2026).
7. Meszaros, G. *xUnit Test Patterns: Refactoring Test Code* / G. Meszaros. – Boston : Addison-Wesley, 2007. – 883 p. – ISBN 978-0-321-20790-9.
8. Google Chrome Developers. *Lighthouse Performance Scoring* / Google // Chrome Developers Documentation [Электронный ресурс]. – Режим доступа: <https://developer.chrome.com/docs/lighthouse/performance/performance-scoring> (дата обращения: 03.03.2026).
9. Martin, R. C. *Clean Architecture: A Craftsman's Guide to Software Structure and Design* / R. C. Martin. – Boston : Prentice Hall, 2017. – 432 p. – ISBN 978-0-13-449416-6.

UDC 004.415.53

REVIEW AND COMPARATIVE ANALYSIS OF UX TEST AUTOMATION METHODS FOR BUSINESS APPLICATIONS

Gugalev A.S.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Scientific advisor: Kosareva E.M. – Assistant Lecturer of the Department of ICSD

Annotation. The article presents a comparative analysis of three methods for automating user experience (UX) quality control in corporate information systems: automated accessibility auditing, visual regression testing, and perceived performance monitoring. Each method is evaluated against three criteria – degree of automation, test stability, and the criticality of detected defects to end-user operations. Based on an analysis of W3C industry standards and leading automation tool documentation, a rational quality assurance strategy for business-oriented software products is substantiated.

Keywords: UX automation, business applications, UI testing, visual testing, WCAG, Lighthouse, user experience.