

УДК 004.056

МЕТОДЫ ОБЕСПЕЧЕНИЯ ОТКАЗОУСТОЙЧИВОСТИ ПРОГРАММНЫХ РОБОТОВ ПРИ ОБНОВЛЕНИЯХ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Анатолий Иванович Бутенко

доктор сельскохозяйственных наук, профессор

but_tolik@mail.ru

Александр Олегович Быстров

магистрант

bystrow1999@yandex.ru

Мичуринский государственный аграрный университет

г. Мичуринск, Россия

Аннотация. В статье рассматривается часто возникающая проблема обеспечения стабильности функционирования RPA-роботов при изменениях пользовательских интерфейсов и корпоративных систем. Анализируются причины возникновения сбоев при обновлениях ПО, а также предлагается классификация методов повышения отказоустойчивости: от архитектурных решений и предиктивной аналитики до самообучающихся систем на основе искусственного интеллекта. Особое внимание уделяется концепции самовосстанавливающихся роботов как перспективному направлению развития RPA-технологий.

Ключевые слова: RPA, отказоустойчивость, самовосстанавливающиеся системы, self-healing, обновления ПО, селекторы, автоматизация.

Сегодня роботизированная автоматизация процессов (Robotic Process Automation, RPA) все чаще применяется различными организациями. Однако по мере масштабирования роботизации приходится сталкиваться с серьезным препятствием — высокой стоимостью поддержки и обслуживания роботов. Именно обслуживание является самым значительным барьером для масштабирования RPA программ.

Главная причина такой ситуации — низкая устойчивость RPA-роботов к обновлениям, заложенная в самой идее технологии [1]. Роботизация обеспечивает интеграцию между системами, имитирует действия пользователя на уровне пользовательского интерфейса. Роботы используют перехват и распознавание изображений с экрана или парсинг дерева структур веб-страниц и приложений. У пользовательских интерфейсов часто изменяются названия полей, перемещаются кнопки, появляются новые всплывающие окна, обновляются макеты страниц. То, что для человека является незначительным изменением, для RPA-робота становится критическим сбоем, требующим ручного вмешательства.

Цель данной статьи — систематизировать существующие и перспективные методы устранения сбоев RPA-роботов при обновлениях программного обеспечения и предложить комплексный подход к повышению отказоустойчивости роботизированных процессов.

Для разработки эффективных методов защиты необходимо понимать причину возникновения сбоев. Анализ литературы и практики внедрения RPA позволяет выделить несколько основных причин:

Изменение селекторов элементов [2]. Селекторы — это уникальные идентификаторы, которые RPA использует для поиска элементов интерфейса. Они могут включать идентификаторы, классы, заголовки, XPath-пути и другие атрибуты. При обновлении приложений эти атрибуты часто меняются и селектор становится недействительным.

Более серьезная проблема — изменение структуры страницы или экрана: перемещение элементов, добавление новых секций, изменение порядка полей в

форме. В таких случаях даже гибкие селекторы могут не найти целевой элемент, так как изменился его контекст.

Появление непредвиденных препятствий. Всплывающие окна, баннеры с согласием на обработку cookies, модальные диалоги обновлений, сообщения о технических работах — эти элементы могут блокировать выполнение сценария, появляясь в непредсказуемые моменты.

Изменения данных и форматов. Обновления затрагивают не только интерфейсы, но и форматы данных. Изменение структуры API-ответов, добавление новых полей в отчетах, модификация форматов дат и чисел могут привести к сбоям в обработке данных.

Таймауты и проблемы производительности. Обновления могут влиять на производительность приложений. То, что раньше загружалось за какой-либо определенную дельту времени, после обновления может требовать совершенно другую. Статические таймауты, заложенные в сценариях, перестают работать корректно.

Теперь рассмотрим архитектурные методы повышения устойчивости. Защита от сбоев должна закладываться еще на этапе проектирования и разработки RPA-решений [3].

Одним из способов избежать ошибок при выборе элементов интерфейса является создания репозитория UI-элементов и повторное использование определенных в нем селекторов.

Вместо того чтобы определять селекторы в каждом сценарии, целесообразно создавать централизованные репозитории UI-элементов. Такой подход позволяет обновлять свойства элементов в одном месте, автоматически применяя изменения во всех процессах, использующих эти элементы. Это особенно важно при масштабировании, когда один и тот же элемент может использоваться в большом количестве роботов.

Другой способ — это иерархическая стратегия селекторов, которая подразумевает определение нескольких вариантов идентификации одного элемента. Если основной селектор не срабатывает, система последовательно

перебирает запасные варианты. Например, сначала предпринимается попытка найти элемент по его уникальному идентификатору, затем — по тексту внутри элемента, после — по позиции в форме. Такой подход значительно повышает вероятность успешного выполнения даже при изменении интерфейса.

Следующим способом повышения устойчивости является использование API (Application Programming Interface). Там, где это возможно, следует отдавать предпочтение API-интеграциям перед UI-автоматизацией. API не зависит от изменений пользовательского интерфейса и обеспечивает более высокую скорость и надежность.

Еще один метод улучшения стабильности – применение стратегии динамического ожидания. Фиксированные таймауты могут вызывать ошибки, если изменится скорость ответа системы. Более эффективны динамические стратегии: ожидание появления определенного элемента, ожидание изменения состояния, интеллектуальные таймауты, адаптирующиеся к текущей производительности приложения.

Перспективным направлением является повышение отказоустойчивости RPA с помощью внедрения искусственного интеллекта и машинного обучения, позволяющих роботам адаптироваться к изменениям без участия человека.

Самовосстановление системы - робот фиксирует ошибку и в реальном времени анализирует ее причину, подбирает другие способы выполнения действия без прерывания работы. Научная архитектура self-healing RPA систем включает несколько ключевых компонентов: сбор телеметрии в реальном времени, интеллектуальную диагностику с классификацией типов отказов, адаптивные механизмы восстановления и механизмы непрерывного обучения на основе обратной связи.

Внедрение самовосстанавливающихся систем позволяет:

- сократить время простоев с нескольких часов/дней до минут;
- снизить затраты на обслуживание;
- повысить масштабируемость.

Рассмотрим функционирование самовосстанавливающихся систем на примере UiPath Healing Agent — решения, доступного на рынке с 2025 года.

При возникновении ошибки агент не просто логирует сбой и останавливается. Он выполняет анализ состояния UI и приложения непосредственно в момент сбоя, стремясь распознать ошибку, причины и альтернативные решения.

Self-healing агент одновременно оценивает несколько стратегий и выбирает наиболее подходящую в зависимости от типа отказа:

-Регенерация селекторов — при изменении атрибутов элемента система анализирует текущую структуру UI и генерирует новые селекторы, сохраняющие ту же целевую функцию, но адаптированные к новым свойствам элемента.

-Семантическое целеполагание — с использованием NLP (Natural Language Processing) система понимает, что изменившаяся метка (например, "Имя" → "Ваше имя") обозначает одно и то же.

-Переключение на компьютерное зрение — если селекторные методы не работают, система переключается на распознавание изображений, используя скриншоты, сохраненные при разработке.

-Управление препятствиями — система распознает всплывающие окна, баннеры и диалоги и применяет соответствующие стратегии их закрытия.

При сложных отказах агент может применять несколько стратегий последовательно: сначала закрыть всплывающее окно, затем скорректировать селектор и, наконец, применить таймаут ожидания загрузки страницы.

Self-healing системы обычно поддерживают два режима работы, что позволяет организациям балансировать между автономностью и контролем:

-Режим самовосстановления — агент автоматически применяет исправления и продолжает выполнение, обеспечивая непрерывность бизнеса.

-Режим рекомендаций — система фиксирует детальную информацию о сбоях и предлагает конкретные исправления, которые разработчик может проверить и утвердить.

Ключевой характеристикой self-healing систем является способность к обучению на основе опыта. Повторяющиеся коррекции автоматически фиксируются и улучшают будущее поведение робота. Например, после нескольких корректировок выбора элемента точность действий возрастает, снижая необходимость дальнейшего вмешательства.

Технологические решения необходимо дополнять организационными мерами, направленными на предотвращение сбоев и минимизацию их последствий [3].

Внедрение тестирования как неотъемлемой части жизненного цикла RPA-роботов позволяет выявлять проблемы до того, как они повлияют на бизнес-операции [4]. Непрерывное выполнение тестов на протяжении всего развертывания роботов обеспечивает их стабильность в продуктивной среде и снижает количество сбоев, вызванных изменениями в автоматизируемых приложениях.

Современные платформы RPA предоставляют средства мониторинга и аналитики, позволяющие отслеживать успешность выполнения, выявлять причины отказов, систематизировать их. Важно измерять не только факт сбоя, но и эффективность восстановления, точность диагностики.

Успешное масштабирование RPA требует четкого разделения ответственности между бизнес-подразделениями и командой разработки\поддержки:

- Владелец процесса отвечает за правила, регламент, критерии качества результата и приоритеты изменений.

- Владелец решения отвечает за работоспособность робота, мониторинг, управление изменениями (обновления интерфейсов, доступы, логирование).

Такое разделение предотвращает конфликты и обеспечивает своевременное реагирование на изменения.

Эффективная практика — заблаговременное информирование команды RPA о планируемых обновлениях корпоративных систем [5].

Таким образом, низкая отказоустойчивость RPA-роботов при обновлениях программного обеспечения является одной из главных проблем, возникающих при масштабировании. Данная проблема решается путем применения следующих методов.

Архитектурные подходы (репозитории элементов, fallback-цепи, API-интеграции) создают базовый уровень устойчивости. Организационные меры (центры компетенций, непрерывное тестирование, мониторинг) обеспечивают процессы предотвращения и быстрого реагирования на сбои.

Сейчас классическое RPA стремится к интеллектуальной автоматизации процессов (IPA) или когнитивному RPA (CRPA). Сочетание RPA с искусственным интеллектом, машинным обучением, обработкой естественного языка и компьютерным зрением создает качественно новый уровень устойчивости и адаптивности.

Развитие attended-автоматизации, когда роботы работают в одном пространстве с человеком, создает новые возможности для повышения устойчивости. Человек может обучать программу, корректируя действия в сложных ситуациях, а робот — запоминать эти коррекции и применять их в будущем.

Наиболее перспективным направлением является внедрение самовосстанавливающихся (self-healing) систем на основе искусственного интеллекта. Такие системы способны в реальном времени анализировать сбои, применять множественные стратегии восстановления и обучаться на основе опыта, трансформируя автоматизацию из статичного, хрупкого инструмента в динамическую, адаптивную capability, создающую все большую ценность с течением времени.

Для российского контекста особую актуальность приобретает развитие отечественных платформ RPA с встроенными self-healing механизмами, способными учитывать специфику локальных корпоративных систем и стандартов.

Список литературы:

1. Mullakara N. Introduction to Robotic Process Automation: a guide to the future of modern business process automation / [S. l.]: Apress, 2023. 232 p.
2. Маштаков М. М., Широкова С. В., Болсуновская М. В. Применение технологии RPA в процессах управления и принятия решений // Technoeconomics: международный журнал. 2023. С. 29–40.
3. Васюкевич Н. Н. Автоматизация бизнес-процессов компании с использованием RPA и искусственного интеллекта // Тенденции экономического развития в XXI веке: материалы VII Междунар. науч.-практ. конф., Минск, 28 февр. 2025 г. / Белорус. гос. ун-т ; редкол.: А. А. Королёва (гл. ред.) и др. Минск: БГУ, 2025. С. 407–409.
4. Пятецкий В. Е., Белых П. В., Елесина К. Е. Роботизация бизнес-процессов (RPA): методические указания / Москва: МИСиС, 2023. 28 с.
5. Ермаков Е. В. Цифровая трансформация: изучение влияния цифровых технологий на бизнес-процессы: монография / Москва: Русайнс, 2024. 64 с. ISBN 978-5-466-08402-3.

UDC 004.056

METHODS FOR ENSURING FAULT TOLERANCE OF SOFTWARE ROBOTS DURING SOFTWARE UPDATES

Anatoly Iv. Butenko

doctor of agricultural sciences, professor

but_tolik@mail.ru

Alexander Ol. Bystrov

master's student

bystrow1999@yandex.ru

Michurinsk State Agrarian University

Michurinsk, Russia

Abstract. The article examines the common problem of ensuring the stable operation of RPA robots under conditions of frequent changes to user interfaces and corporate systems. The causes of failures during software updates are analyzed, and a classification of methods for increasing fault tolerance is proposed: from architectural solutions and predictive analytics to self-learning systems based on artificial intelligence. Special attention is paid to the concept of "self-healing" robots as a promising direction in the development of RPA technologies.

Keywords: RPA, fault tolerance, self-healing systems, self-healing, software updates, selectors, automation.

Статья поступила в редакцию 25.02.2026; одобрена после рецензирования 20.03.2026; принята к публикации 31.03.2026.

The article was submitted 25.02.2026; approved after reviewing 20.03.2026; accepted for publication 31.03.2026.