



Битва Low-code и нативного программирования: RPA против Python — эволюция разработки или война подходов

Март 13, 2025

Сергей Вотяков

GR директор PIX Robotics, член Правления RUSSOFT, ректор NARPA
info@narpa.ru | www.narpa.ru

Введение

Современная разработка программного обеспечения и автоматизация процессов сталкиваются с двумя основными подходами: Low-code платформы, такие как RPA (Robotic Process Automation), и традиционное нативное программирование, например, на Python. Каждый из этих подходов имеет свои преимущества и ограничения, что порождает дискуссии о том, какой из них более эффективен. В данном докладе мы рассмотрим ключевые различия между Low-code и нативным программированием, сравним RPA и Python, а также обсудим будущее автоматизации.



Low-code vs нативное программирование: ключевые различия

- **Скорость разработки:** Low-code платформы позволяют создавать решения быстрее благодаря готовым компонентам и визуальному интерфейсу.
- **Гибкость:** Нативное программирование предоставляет больше возможностей для кастомизации и решения сложных задач.
- **Порог входа:** Low-code требует меньше времени на обучение, что делает его доступным для широкого круга пользователей.

RPA

Преимущества

- Скорость разработки: Создание RPA-робота занимает значительно меньше времени по сравнению с написанием аналогичного решения на Python.
- Низкий порог входа: Обучение RPA занимает около месяца, что позволяет быстро подготовить специалистов.
- Самодостаточность: RPA платформы не требуют подключения внешних библиотек, что упрощает процесс разработки.

Ограничения:

- Ограниченная гибкость: RPA менее подходит для сложных задач, требующих глубокой кастомизации.
- Зависимость от платформы: Решения, созданные на RPA, могут быть привязаны к конкретной платформе.

Python

Преимущества

- Универсальность: Python подходит для решения широкого спектра задач, от простых скриптов до сложных приложений.
- Гибкость: Возможность интеграции с другими технологиями и библиотеками делает Python мощным инструментом.
- Сообщество: Большое количество документации и активное сообщество разработчиков.

Ограничения

- Время разработки: Создание решений на Python может занимать больше времени, особенно для сложных задач.
- Порог входа: Обучение Python требует значительных временных затрат (около года для достижения среднего уровня).

Заключение

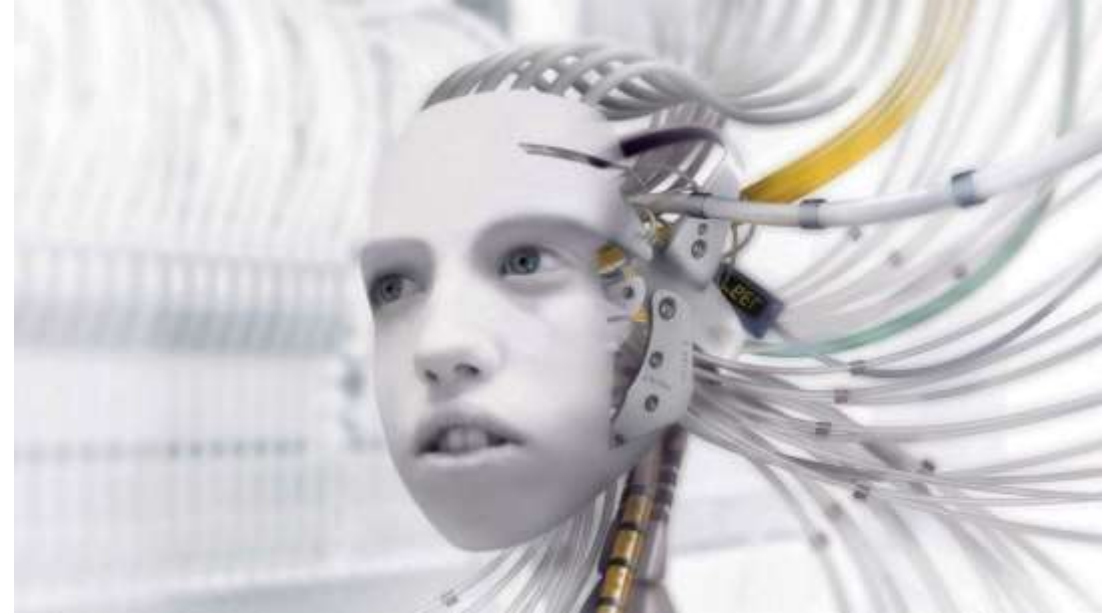
Битва между Low-code и нативным программированием — это не война, а эволюция подходов к разработке. RPA и Python имеют свои сильные стороны, и их интеграция открывает новые возможности для автоматизации. В будущем мы увидим больше гибридных решений, где скорость разработки RPA будет сочетаться с гибкостью Python, что позволит создавать более эффективные и универсальные системы.

Возможности генеративного ИИ

Автоматизация и генерация кода: LLM способны автоматически генерировать код на Python, используя естественные языковые запросы.

Понимание контекста и синтаксиса: Модели понимают контекст задачи и синтаксис Python.

Поиск и исправление ошибок: LLM могут анализировать существующий код, находить ошибки.



Заключение 2

С учетом того, что RPA часто требует вставок нативного кода для автоматизации сложных процессов, выбор между RPA и Python становится менее важным. Вместо этого вы можете использовать гибридный подход, который сочетает скорость и простоту RPA с гибкостью и мощностью нативного программирования. LLM играют ключевую роль в этом процессе, помогая обучать персонал, генерировать код и поддерживать разработку.

Рейтинг Платформ RPA 2025 – уже на портале RPA2.ru

1 место - PIX RPA



Лидер по стабильности и скорости развития, обладает самой большой клиентской базой. Выстроил регулярную работу с учебными заведениями, что видно по анонсам вузов о запусках учебных программ, хакатонах и мастер-классов по работе на PIX RPA.

Большой фокус на инновации: развивает функциональность на Linux и Windows, сотрудничает с отечественными и зарубежными разработчиками GPT для ускорения внедрения технологий ИИ (AI). Вендор также разрабатывает инструменты для анализа бизнес-процессов и эффективности, что позволяет применять RPA шире и делает платформу самой подходящей для масштабной роботизации.

Для погружения новичков вендор выпускает курсы, но есть ограничения в пробной версии на количество шагов. Обладает самым большим и активным русскоязычным RPA-сообществом в Телеграм.

2 место - Primo RPA



Primo RPA создан по образу и подобию UiPath, что на старте было выигрышной позицией, однако привело к функциональным ограничениям и не позволило развиваться с той скоростью, что и конкуренты. В 2023 году переняли идею UiPath – инструмент для сбора инициатив по роботизации Idea Hub, который, в отличие от оригинала, не позволяет полностью автоматизировать управление бизнес-процессами.

Primo RPA не спешит раскрывать кейсы внедрения платформы: в библиотеке TAdviser нет ни одного кейса, но на сайте вендора описаны семь. Сообщество пользователей в официальной группе Телеграм сильно отстает от других RPA-платформ.

3 место - Robin



No-code подход, выбранный Robin, прост в использовании и подходит для начинающих пользователей, которые хотят впервые попробовать

RPA^{2.0}

Robotic Process Automation

Бесплатный образовательный проект PIX triathlon

1. Создание ИИ агентов/программных роботов
2. Создание дашбордов в BI системе
3. Визуализация бизнес-процессов
4. Применение ИИ в т ч во фреймворках разработки универсальных алгоритмов
5. Использование экосистемы для решения задач



Образовательные программы НАРПА

В настоящее время, выпускники Академии стали основой центров RPA компетенций в таких компаниях как

РОСАТОМ, РЖД, АЛРОСА, ВТБ, Банк Открытие, ГПБ, МТС Банк, РосВодоКанал, НИПИГАЗ, Гринатом, ОАК, Уралэнергосбыт, Счетная Палата РФ, Россконгресс, Русал, БорисХоф, Роснефть, Норникель, Спортмастер, ВШЭ, СПбПУ, ТПУ, ПГУПС, УИ, СГУ и другие.



Robotic Process Automation



Спасибо за внимание!

Март 13, 2025

Сергей Вотяков

GR директор PIX Robotics, член Правления RUSSOFT, ректор NARPA
info@narpa.ru | www.narpa.ru