

вебинар

Платформы RPA: как сделать правильный выбор

на примере российских и зарубежных продуктов

На связи: news@mail.ifellow.ru



с **2015**

консалтинг, системная интеграция, тестирование,
комплексная разработка и поставки в области
информационных технологий

600+

квалифицированных
специалистов

образовательная
ЛИЦЕНЗИЯ

№040538 от 25 декабря 2019 г

сертификат

по стандарту ISO 9001:2015 высокий
уровень управления процессами



■ Места присутствия

■ Офисы компании

Вендоры



Дистрибьюторы



Цели внедрения RPA (1/2).

Цели внедрения	Заказчик/основной бенефициар	Источник выгоды	Как часто встречается
Оптимизация/автоматизация рутинных процессов	Кто угодно, чаще всего бэк-офис	Экономия на трудоресурсе, снижение рисков ошибки	Очень часто
Перенос данных при миграциях, дублирование данных для архива	ИТ, команды по миграции, реже — бизнес	Экономия на трудоресурсе (для ручного переноса)	Часто
Замена неконтролируемых скриптов/макросов	Информационная безопасность	Устранения рисков использования «ничейного» кода	Крайне редко, в РФ не встречается

Цели внедрения RPA (2/2).

Цели внедрения	Альтернативные решения	Плюсы RPA	Недостатки RPA
Оптимизация/автоматизация рутинных процессов	<ul style="list-style-type: none">Скрипты, макросы,Доработки информационных систем	<ul style="list-style-type: none">Простота, единообразииПотенциально дешевле и быстрее	<ul style="list-style-type: none">Новый инструментБолее ресурсоемко в долгосрочной перспективе
Перенос данных при миграциях, дублирование данных для архива.	Механизмы прямого переноса данных, интеграция систем.	Простота, стоимость	Зависимость от GUI
Замена неконтролируемых скриптов/макросов.	Хранилище макросов, контролируемое в ручном режиме.	Прозрачный контроль за созданием и использованием кода	Стоит денег, в отличие от «бесплатных» скриптов и макросов

Требования (1/2). Парадигма кода

Цели внедрения	Простота/скорость разработки	Простота поддержки	Code, low code и no code
Оптимизация/автоматизация рутинных процессов	Требуется высокая скорость разработки и доступность инструмента для специалиста невысокой ИТ-квалификации	Требуется возможность гибкого управления большим количеством роботов. Желателен интерфейс для бизнес-пользователя	Можно и без кода
Перенос данных при миграциях, дублирование для архива	Простота роботизации пользовательского интерфейса конкретной ИС	Не критично	Как правило, чистый no code не подходит
Замена неконтролируемых скриптов/макросов	Требуется простота «обёртывания» кода средствами платформы. «Разработка» ведётся кодом	Необходим инструмент, позволяющий контроль большого объёма (~тысячи) решений	Code preferred

Требования (2/2)

Цели внедрения	Возможность масштабирования	Надёжность и безопасность	Ограничения технологий
Оптимизация/автоматизация рутинных процессов	Требуется поддержка масштабирования, как горизонтального (количество), так и вертикального (новые инструменты)	Требования к надёжности и безопасности имеют приоритет ниже, чем скорость разработки и простота использования решения	Работа на пользовательской ОС (и это Windows)
Перенос данных при миграциях, дублирование для архива	Не требуется	Требуется «нулевое» количество ошибок. Полное логирование операций	Отличаются от проекта к проекту
Замена неконтролируемых скриптов/макросов	Требуется поддержка горизонтального масштабирования, в первую очередь для БД хранения логов и репозитория решений	Требуется высокая надёжность контроля, в том числе подпись выложенных сборок ЭЦП	Желательна возможность запуска на разных ОС

Стоимость владения платформой RPA

1. **Стоимость 2-й линии поддержки.** Более высокая стоимость поддержки 2-й линии из-за необходимости дорабатывать роботов в процессе их эксплуатации.
2. **Стоимость лицензий** и инфраструктуры. Требования к железу, как правило, идентичные, софт используется разный.
3. **Стоимость резервирования.** Необходим разумный баланс стоимость/требования. Хорошая практика резервировать только СУБД и сервера управления (если они есть).
4. Возможно стоимости инфраструктуры за счёт использования **контейнеризации** и/или **облачной инфраструктуры**, что на практике встречается очень редко.

Причины большого Time-to-market роботизированных решений

Главная причина большого времени «выхода в свет» — отсутствие конвенции роботизации. То есть, отсутствие предварительной договорённости между заказчиком, ИТ и ИБ.

Анализ + ТЗ

- Не выделен аналитик на поиск процессов
- Пассивный «заказчик»
- Нет автоматизированных инструментов

Разработка

- Низкая скорость получения доступов
- Отличия среды разработки от продакшена

Тестирование

- Отсутствие встроенных инструментов тестирования

Возможные архитектурные подходы (1/2)

Код или не код?

Код нужен для:

- Быстрого обращения к БД
- REST API
- Использование широкого спектра структур/объектов в решении
- Непосредственного выполнения скриптов любого рода
- Использование принципов ООП при написании отдельных модулей платформы

Можно без кода:

- Работа с TUI/GUI
- Работа со стандартными офисными приложениями
- Работа с браузерным клиентом
- Запуска и остановки программ уже «подготовленных» для нас



Возможные архитектурные подходы (2/2) Простота или экосистема?

Что может отдельный робот:

- Локальное исполнение заданного алгоритма с использованием всех доступных инструментов

А что ещё может платформа:

- Запуск роботов по расписанию и/или по событию
- Централизованное обращение к внешним сервисам, в том числе ML
- Удобные распределение нагрузки и мониторинг роботов

Функционал средств разработки

Точно понадобится:

- Работа с браузерными приложениями через Chrome и Edge
- Распознавание текста
- Работа с MS Office, работа с почтой, чтение pdf без использования GUI
- Вызов кода (консоль, макросы)
- Работа с любым GUI через селекторы



Возможно понадобится:

- Работа с браузерными приложениями (Mozilla, Opera etc)
- Распознавание картинок
- Работа с аналогами MS Office («Мой Офис») без использования GUI
- Прямая работа с кодом
- Средство работы с SQL, REST API



Практические вопросы организации БД

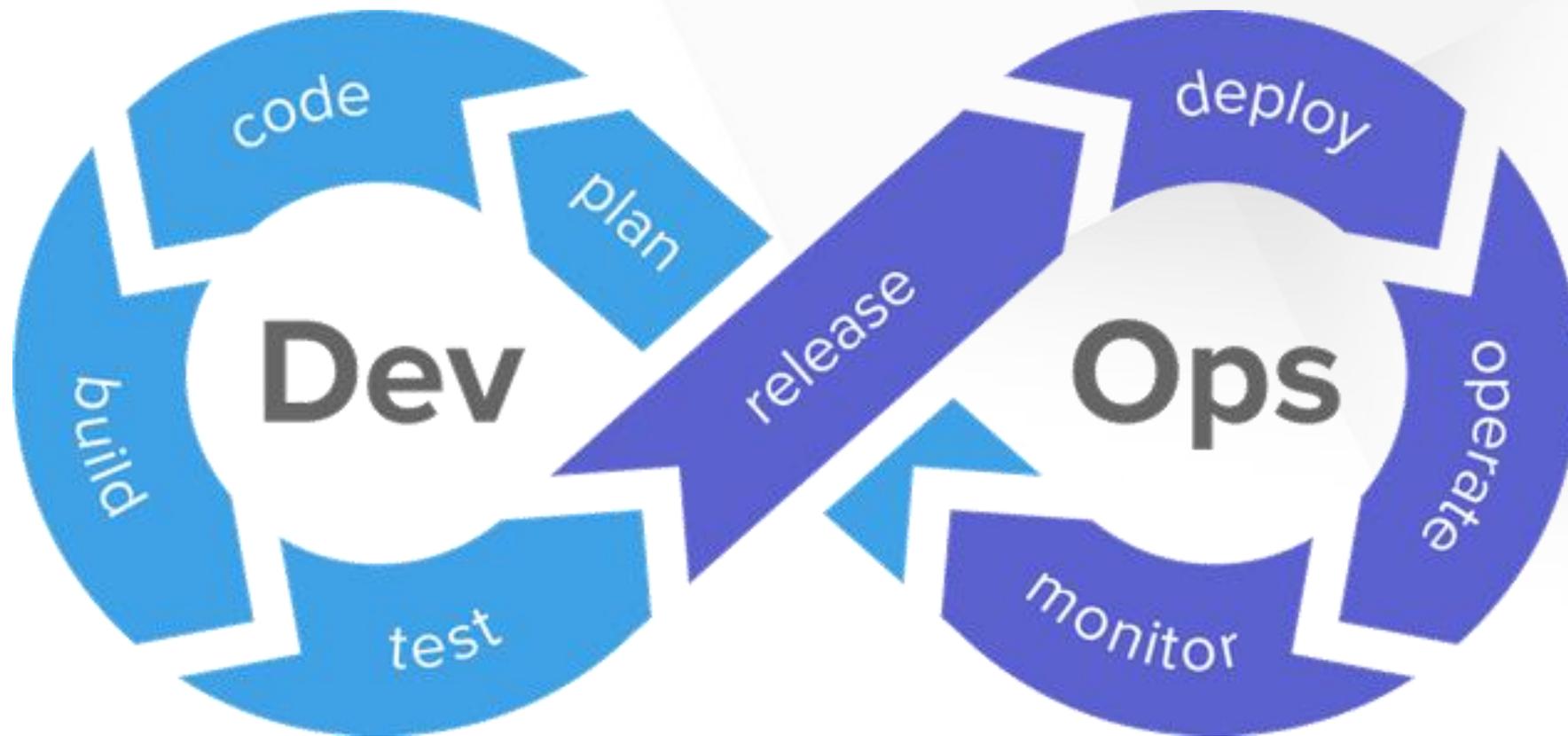
1. Надёжность и производительность служебной СУБД. Для технология RPA в подавляющем большинстве случаев достаточно стандартной функциональности
2. Хранение и анализ информации (логов) за значительный исторический период. Лучшим решением будет регулярное архивирование БД и использование дополнительных инструментов такого класса как Elasticsearch и Grafana
3. Безопасность данных. Доступы, шифрование, обезличивание данных
4. Выделение прикладной БД для использования роботами

RPA и Agile

Принципы Agile в полной мере применимы для RPA:

- Наивысший приоритет — это удовлетворение заказчика
- Изменение требований допустимо на всём протяжении разработки
- Поставка работающего программного обеспечения asap, лучшие практики — это несколько недель
- Частое, вплоть до ежедневного, общение разработчика и заказчика

RPA и CI/CD



Интересное о чем не успели

- **Ролевая модель.** Разработка-тест-пром, какие роли у пользователей, что позволено разработчику
- **Варианты выхода в WWW из закрытого контура.** Проксирование, подключение через фермы виртуализации и другие решения
- **Реальные возможности графического распознавания.** Что могут встроенные движки, когда имеет смысл ML или привлечения коммерческих (более продвинутых) движков
- **Проекты по переходу между платформами**

Хотите обсудить? Пишите нам!
news@mail.ifellow.ru

вебинар

Платформы RPA: как сделать правильный выбор

Вопросы?

На связи: news@mail.ifellow.ru

