

Инструментарий разработчика как сервис:

парадокс автоматизации



DELLEMC

О нас



АНТОН РЫМКУС

 [@rymkus](https://twitter.com/rymkus)



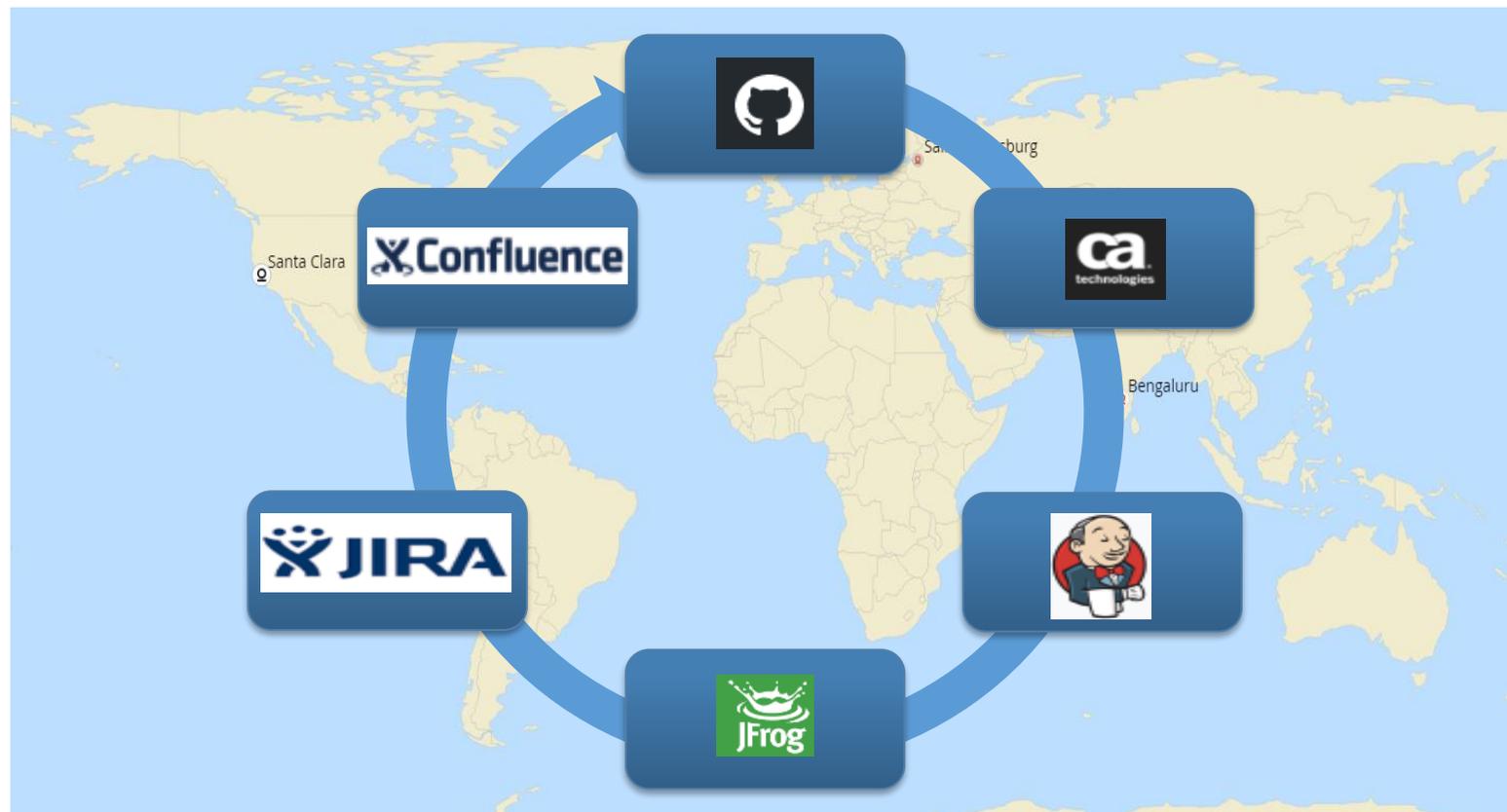
Оксана Некрасова

 [@Oks_Nekrasova](https://twitter.com/Oks_Nekrasova)

Два года назад

Клиенты и сервис

- 3000+ внутренних клиентов по всему миру
- Разные сервисы
- 20+ инстансов тулов в нескольких локациях



Команда поддержки

- 7 админов
- Ручные операции
- Нет документации

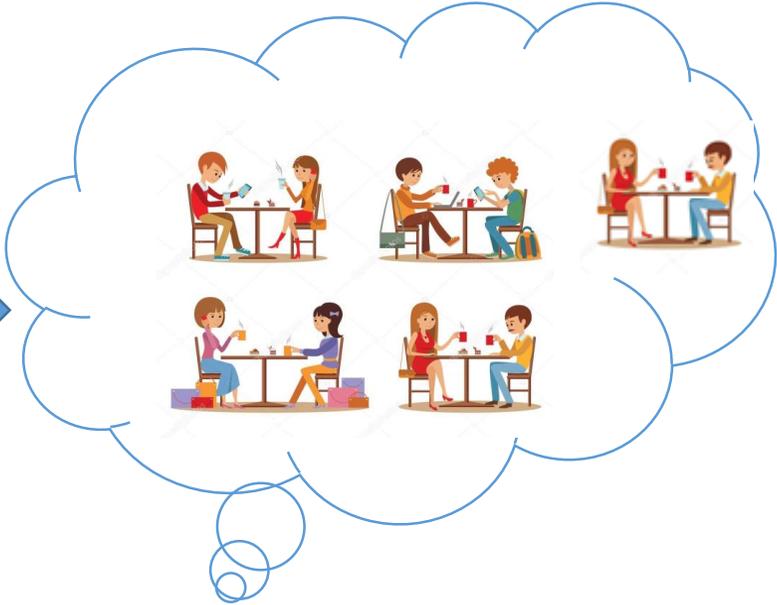
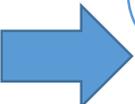
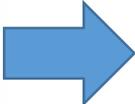


Проблемы

- Ограниченная доступность каждого админа
- Отсутствие поддержки 24/7
- Непрозрачность коммуникаций с клиентами
- Продуктовая система настраивается руками,
- Отсутствие документации
- Отсутствие прозрачности поддержки
- Ручные работы, по-разному у разных людей

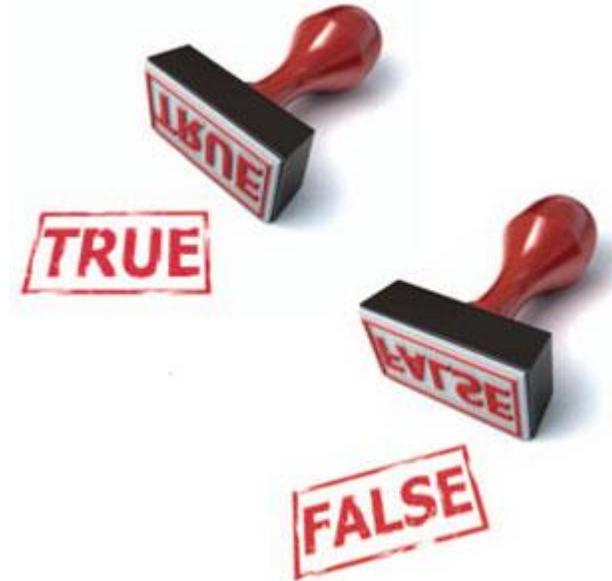


План: создать команду автоматизации



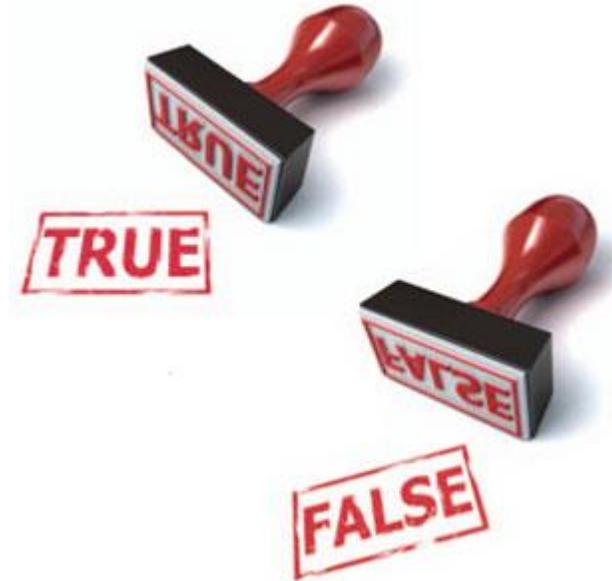
Исходные предположения

Автоматизация не потребует много ресурсов на её поддержку



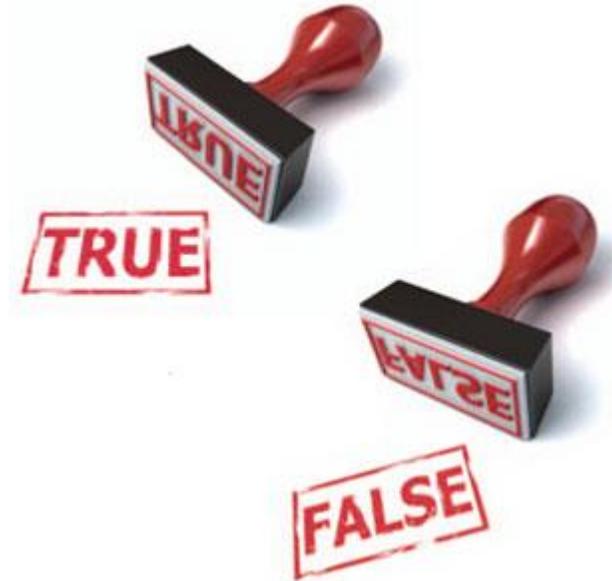
Исходные предположения

Подход с автоматизацией сэкономит много денег



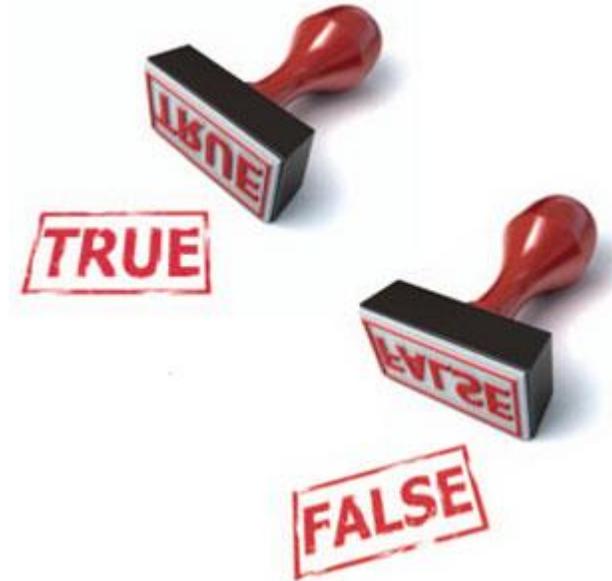
Исходные предположения

Автоматизация всегда предпочтительнее ручной работы

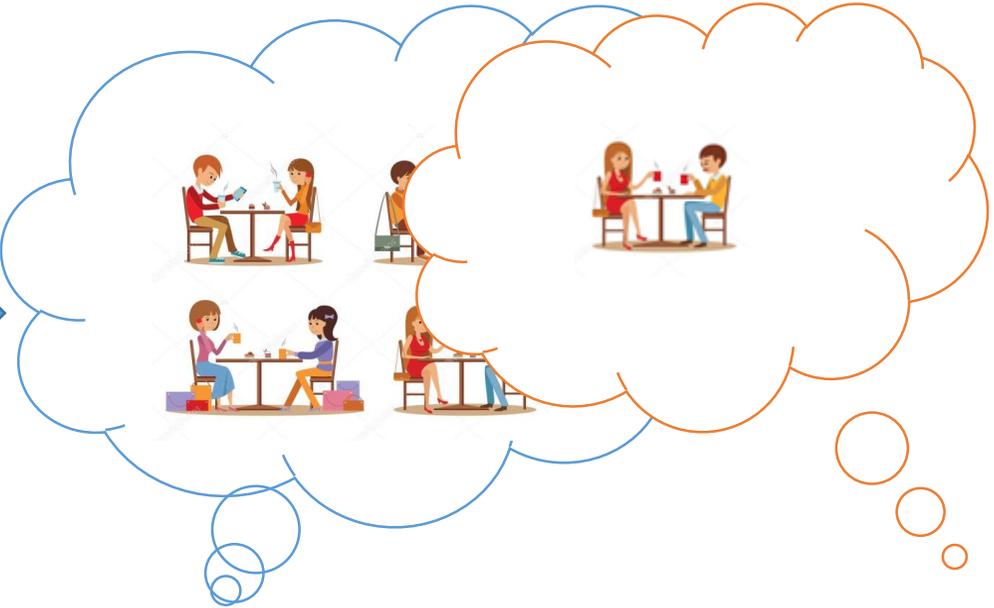
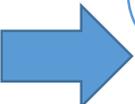
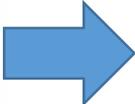


Исходные предположения

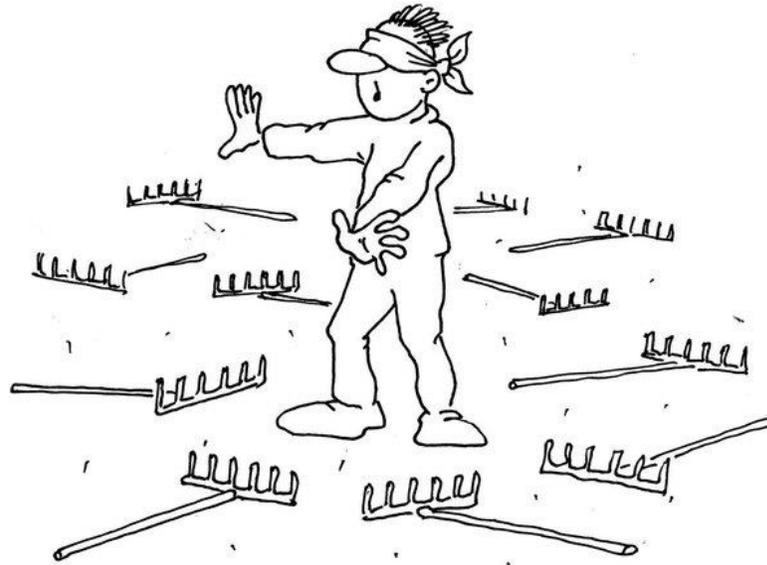
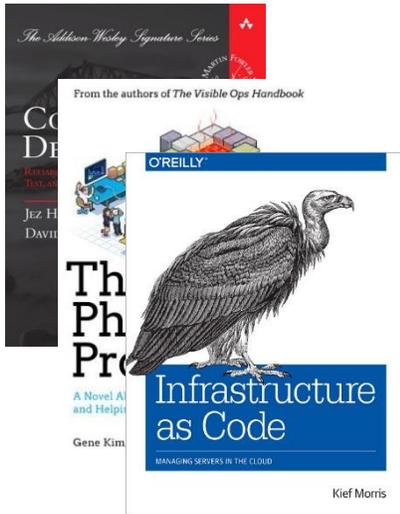
Работы по автоматизации возможно полностью завершить в ограниченные сроки



Мечта: урезать команду автоматизации



Начнем?



Нестабильность коммуникации с заказчиком

Решение



Объединение команды автоматизаторов с командой поддержки



Унифицированные списки рассылок для всех сервисов, система заявок



Общее хранилище документации



Slack канал для команды для ускорения общения

Нестабильность коммуникации с заказчиком

Плюсы:

- Простая схема коммуникации с заказчиками
- Удобная коммуникация внутри команды,
взаимозаменяемость
- Покрытие всех временных зон

Расходы:

- Бесплатно

Infrastructure as a code

Решение: превращение админов в команду разработчиков с поддержкой всех процессов



Infrastructure as a code

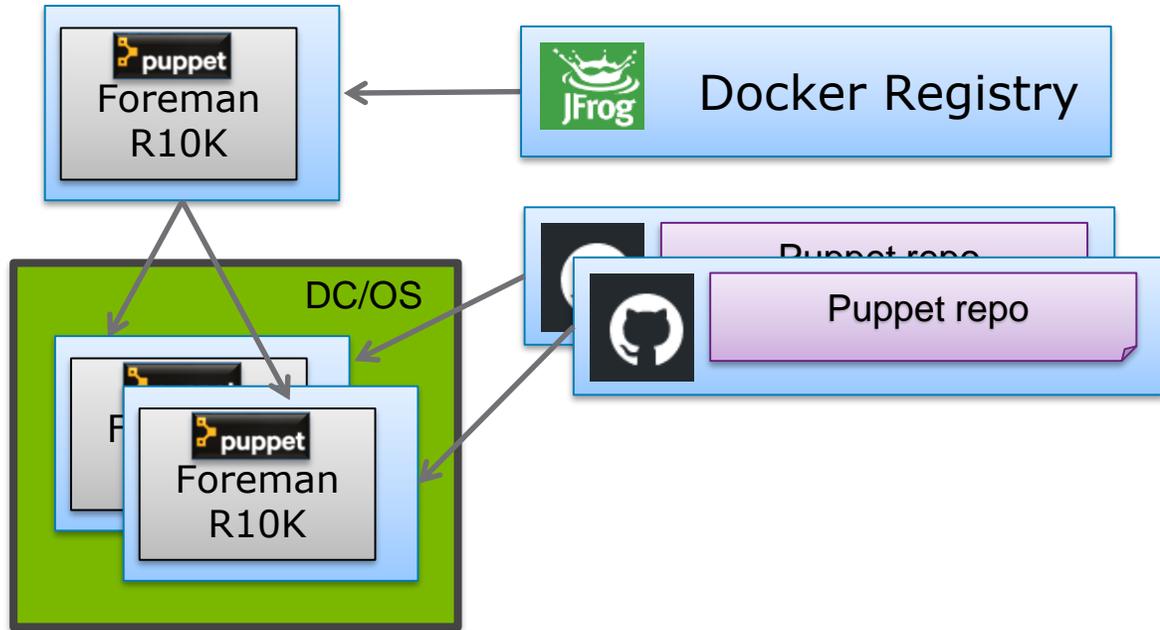
Плюсы:

- Конфигурация как продукт
- Eat your own Dogfood

Расходы:

- Лицензии (GitHub, Artifactory, Jira, Confluence, и т.д.)
- Ресурсы (виртуальные машины, дисковое пространство)

Автоматизация изменений конфигураций



Решение:

- Open Source Puppet
- Интеграция с GitHub (каждый бранч – environment)
- Custom facts

Автоматизация изменений конфигураций

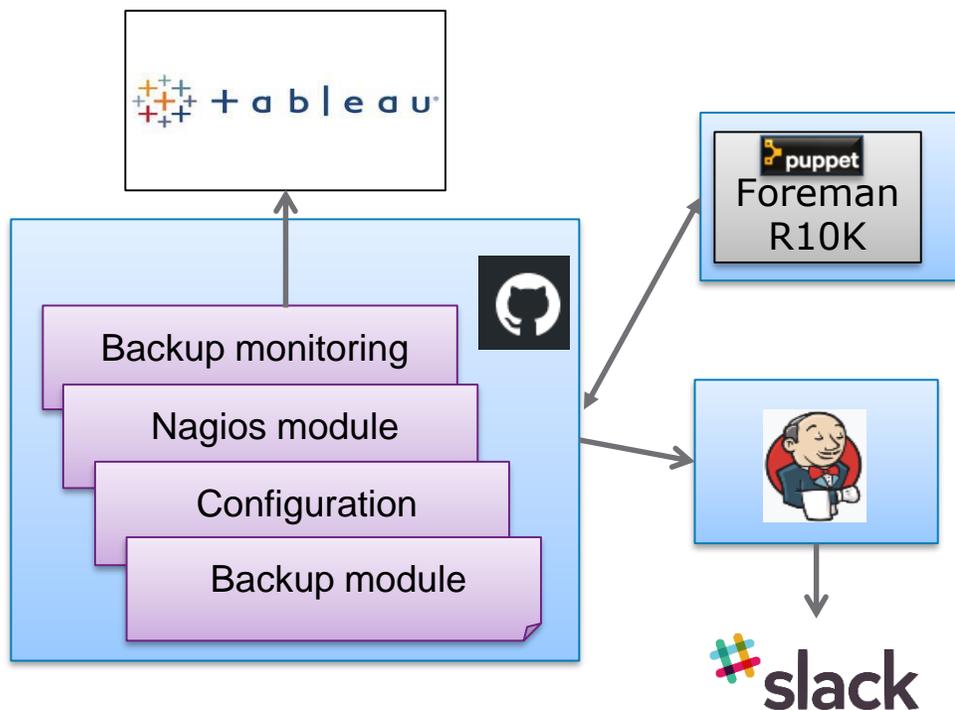
Плюсы:

- Масштабирование
- Гибкость управления

Расходы:

- Инфраструктура
- Расходы на разработку
- Расходы на поддержку

Изменение конфигураций – реализация



Решение:

- Независимые модули для всех процессов
- Визуальное отображение истории состояний
- Последовательность изменений в slack channel

Изменение конфигураций – реализация

Плюсы:

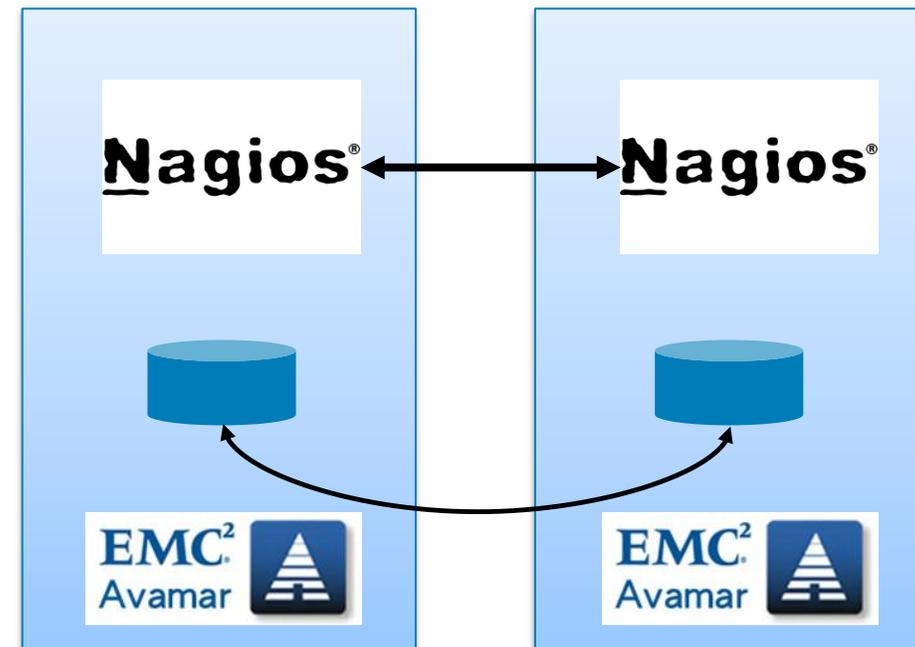
- конфигурация документирована
- изменения отслеживаются и применяются отдельно к каждому инстансу
- переиспользование модулей

Расходы:

- постоянные улучшения
- поддержка модулей, CI/CD и инфраструктуры

Отслеживание состояния системы и восстановление

Решение: Nagios для мониторинга, автоматизация для бэкапов: 2 типа хранения – для быстрого восстановления и песочницы



Отслеживание состояния системы и восстановление

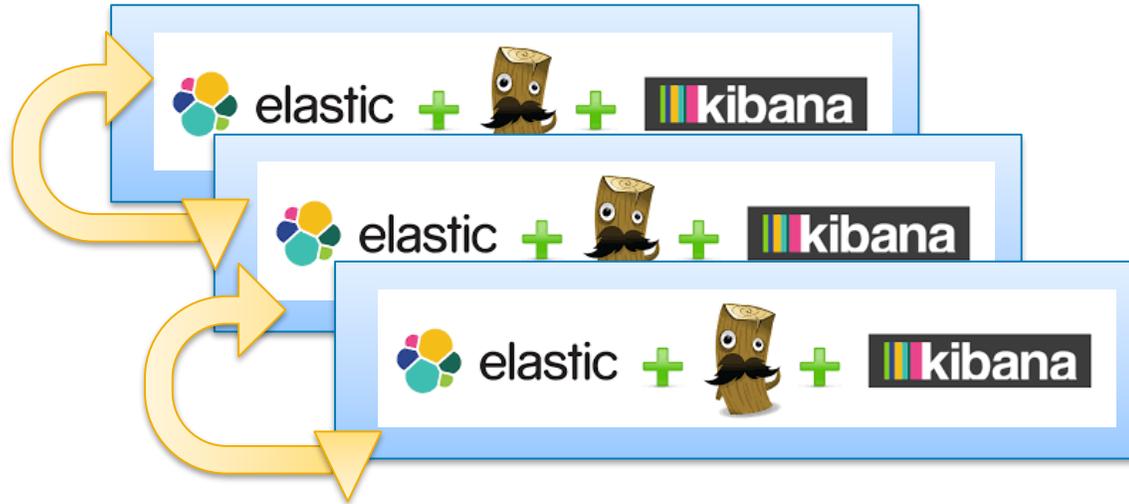
Плюсы:

- Раннее обнаружение проблем
- Анализ и геолокация
- Восстановление данных

Расходы:

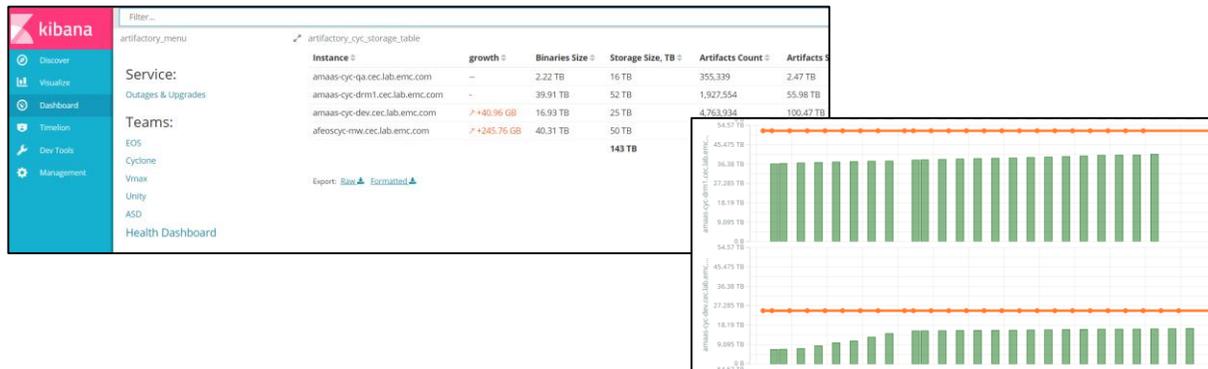
- Лицензии Nagios
- Место под хранение резервных копий
- Инфраструктура
- Поддержка скриптов

Логирование и метрики



Решение:

- ELK stack



Логирование и метрики

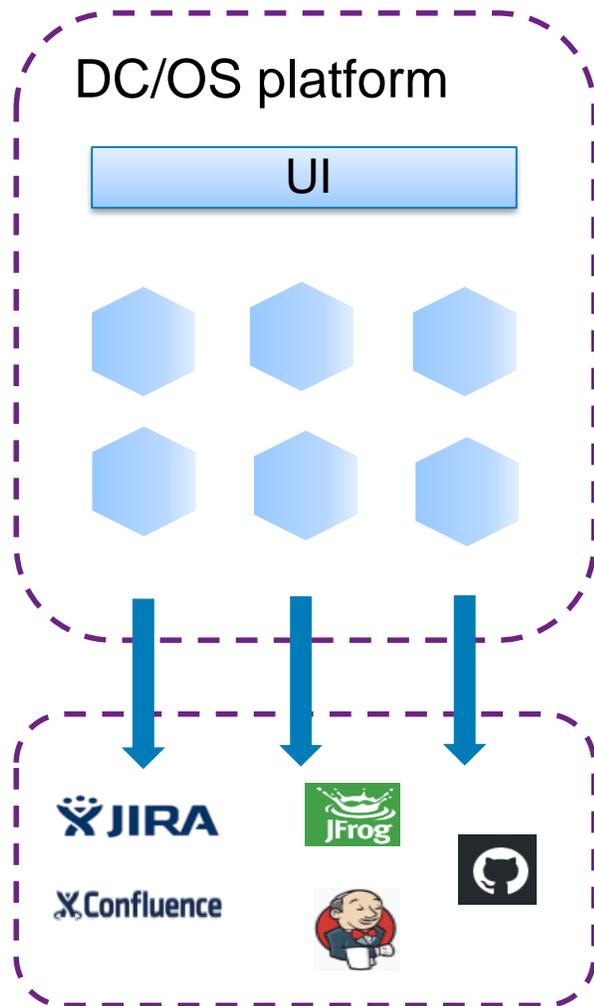
Плюсы:

- Единая точка входа для поиска логов
- Графики для опс-ов и менеджмента
- Оповещения и интеграция с другими тулами
- Легко расширить объем и функционал

Расходы:

- Инфраструктура
- Поддержка
- Непрерывная реализация «хотелок»

Поддержка пользователей



Решение: Автоматизация поддержки на базе микросервисной архитектуры (на PaaS)
Написание API & UI для рутинных операций

Поддержка пользователей

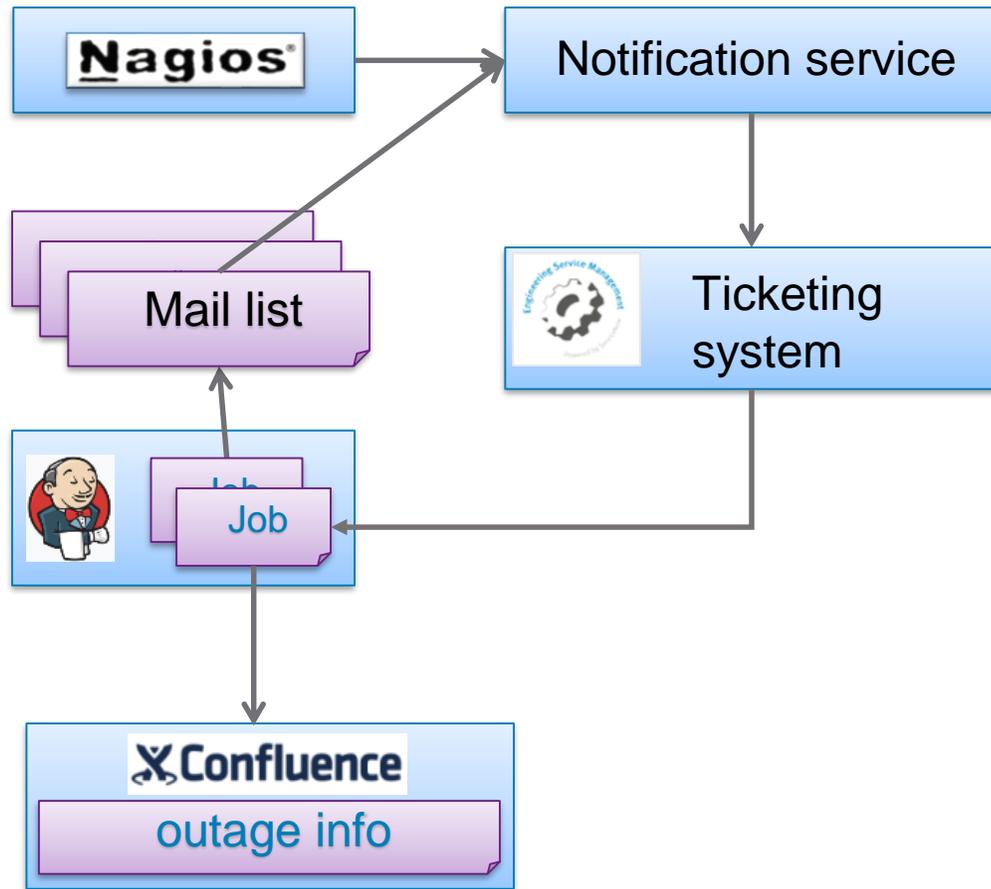
Плюсы:

- Единообразные автоматизированные процедуры для админских задач
- легко передать внешним командам

Расходы:

- Толстая инфраструктура (PaaS) и её поддержка
- Java Spring и UI разработка (нужна дополнительная экспертиза)
- Разработчики смотрят налево и идут в разработку

Сервис оповещений заказчика



Решение: автоматизация оповещений

- Алерты создаются Nagios
- Автообновление списков рассылок
- Возможность подписаться на события
- Автообновление событий
- Отслеживание нескольких локаций

Сервис оповещений заказчика

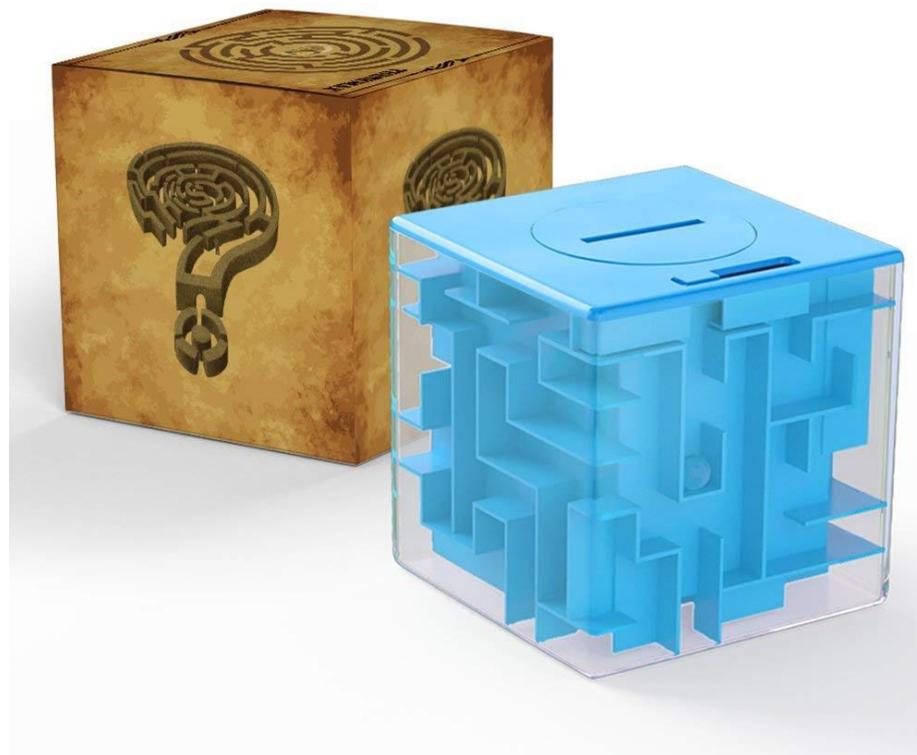
Плюсы:

- Открытость сервиса
- Оповещения о всех событиях
- Ручная подписка

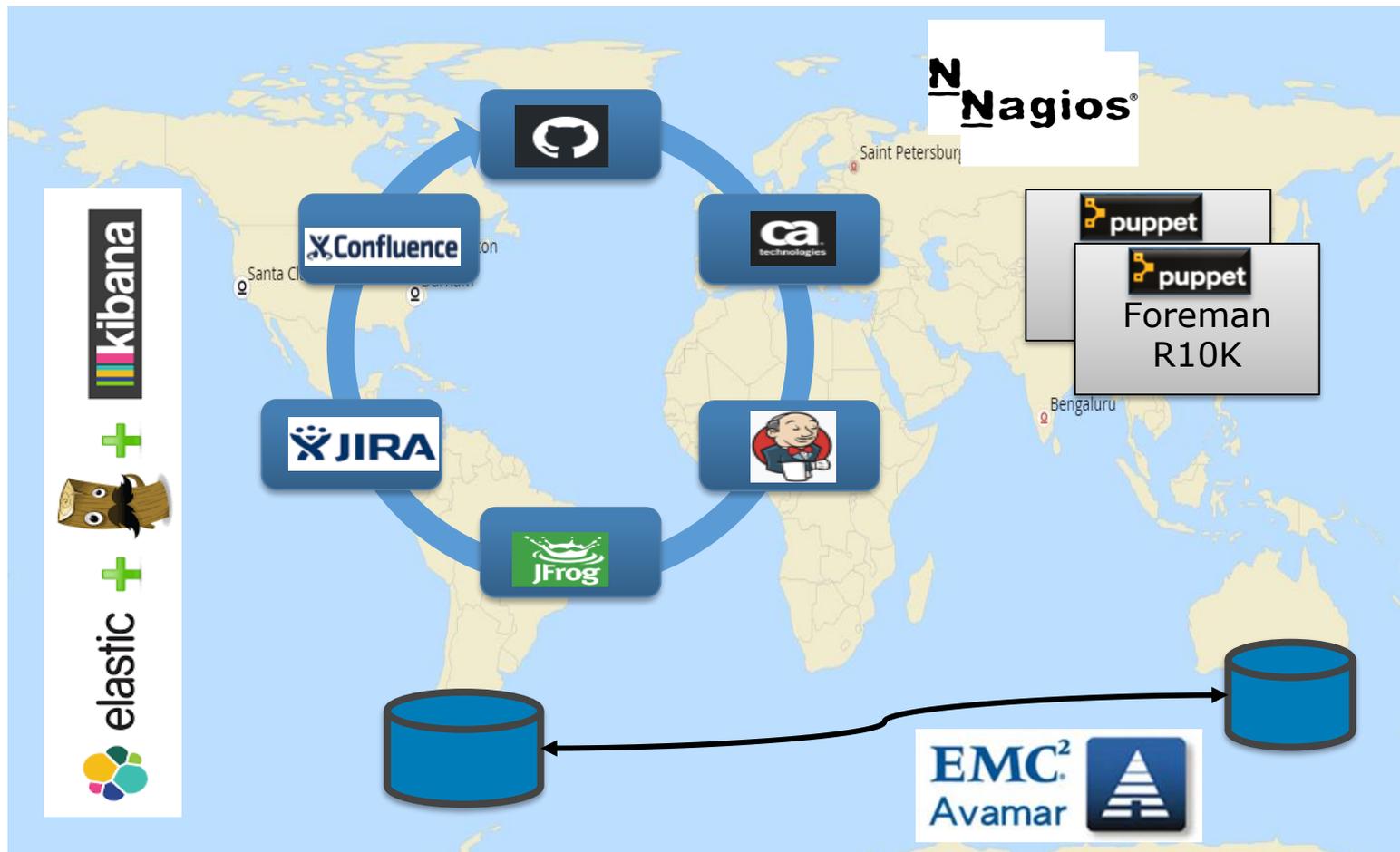
Расходы:

- Разработка
- Инфраструктура
- Поддержка

Что же мы создали?



Результат

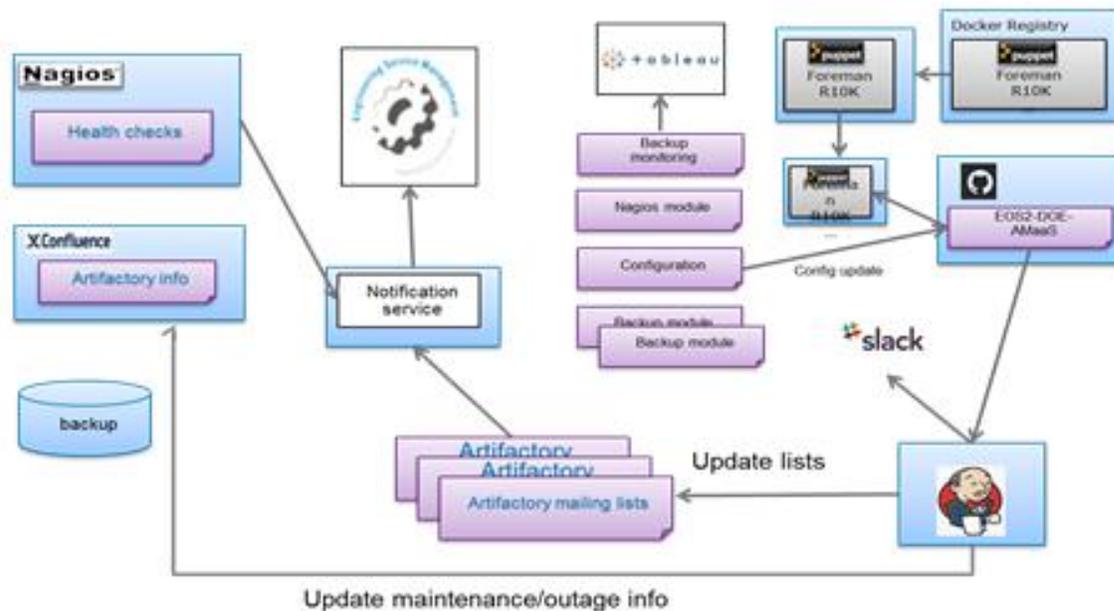
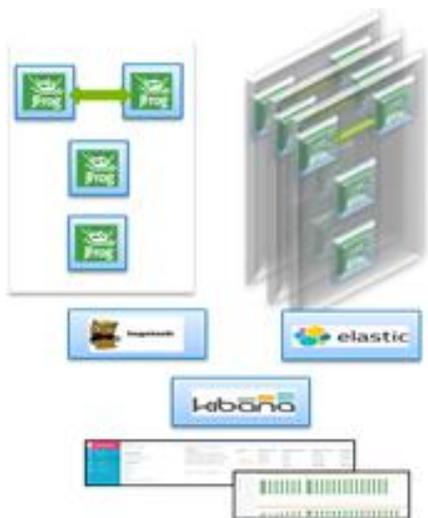


Автоматизировано

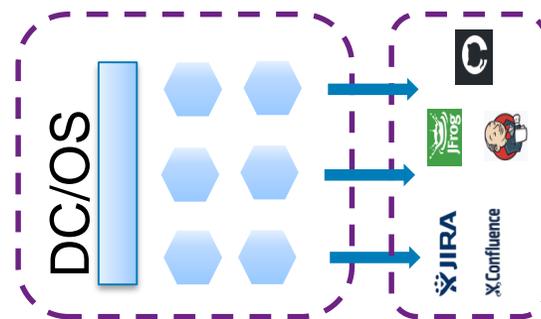
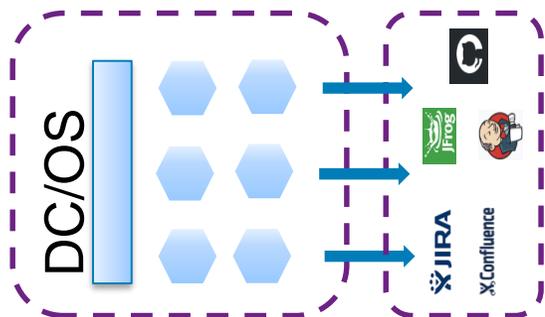
Надежно

Масштабируемо

Результат



Дорого
Сложно
Порог вхождения



Что же мы создали?



Предположения



Автоматизация не потребует много ресурсов на её поддержку

Подход с автоматизацией сэкономит много денег

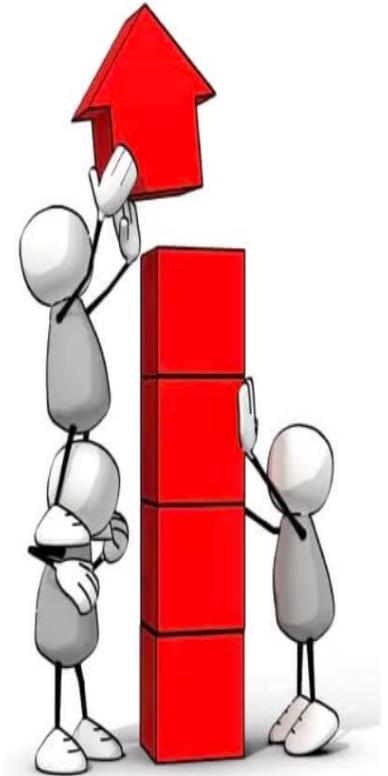
Автоматизация всегда предпочтительнее ручной работы

Работы по автоматизации возможно полностью завершить в ограниченные сроки

Автоматизация не потребует много ресурсов на её поддержку

- 1) Новые версии API тулов появляются часто и регулярно не имеют обратной совместимости
- 2) Продакшн замониторили, но «кто будет контролировать контроллеров?»
- 3) SPoF или распределенная система?

FALSE



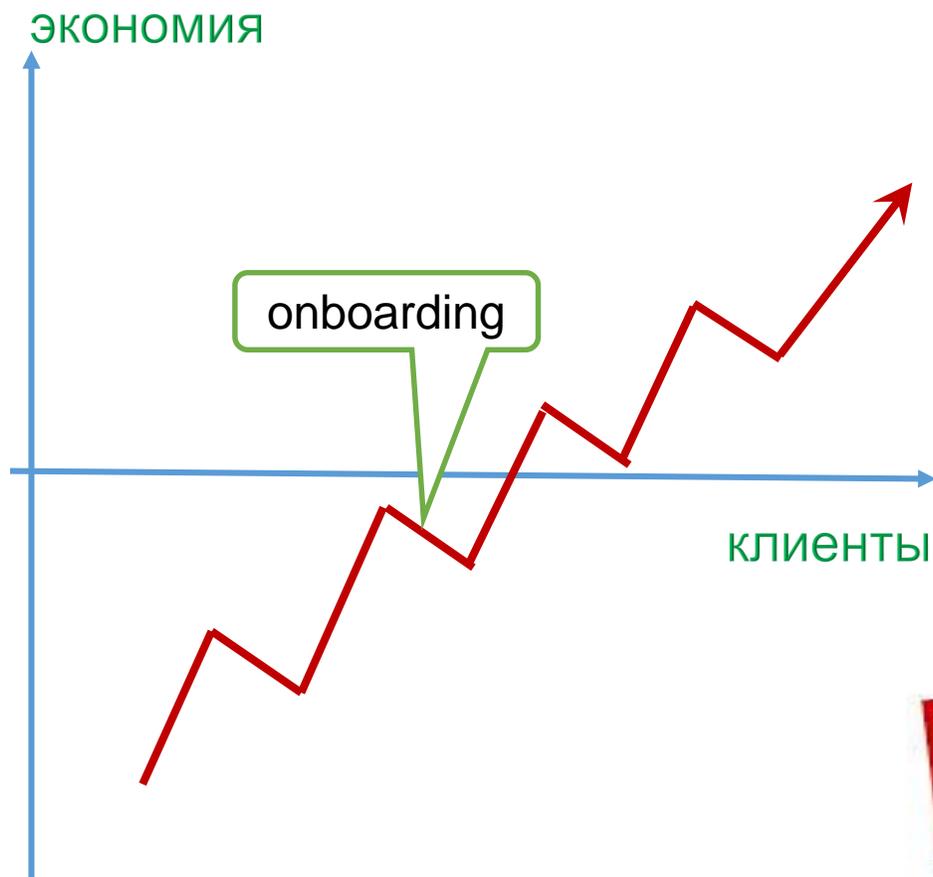
Подход с автоматизацией сэкономит много денег

	Разработчики	Клиенты	instances	Инфраструктура*
До	7	> 3000	20	30 TB P.K.** + 20 VMs
Год спустя	7 + 7	> 6000	50	60 TB P.K. + 80 VMs + 14 лицензий GitHub, Jira, Confluence...
Два года спустя	7 + 7	> 9000	70	80 TB P.K. + 100 VMs + 14 лицензий GitHub, Jira, Confluence...

* – цифры вымысел, но основаны на реальности.

** – P.K. = Резервная Копия

Подход с автоматизацией сэкономит много денег



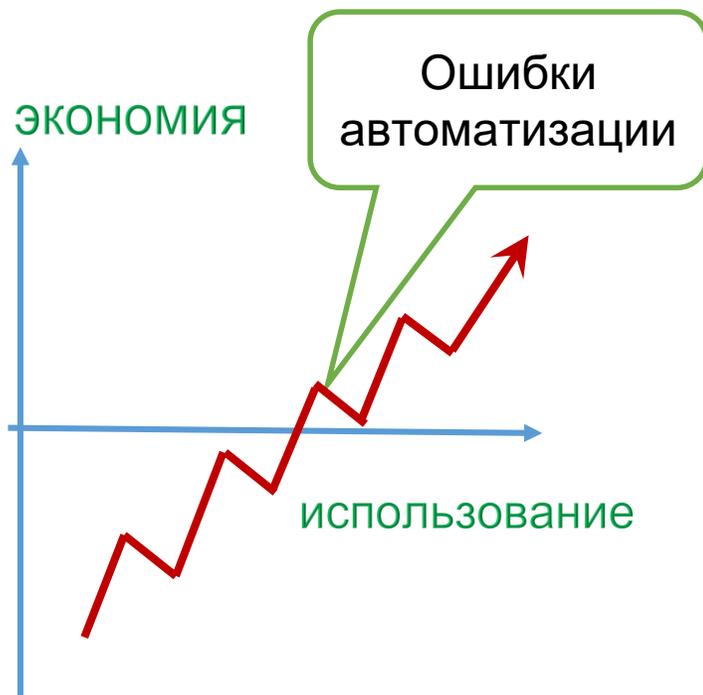
Экономия:

- Лицензирование
- Предотвращение инцидентов
- Бюджет клиентских DevOps/support команд

Автоматизация всегда предпочтительнее ручной работы



Автоматизация всегда предпочтительнее ручной работы



- До применения автоматизации:
 - Разберитесь в процессе, который собираетесь автоматизировать
 - Сравните затраты и выгоду от результата
- Автоматизируйте болевые точки
- Автоматизируйте операции команд приносящих наибольшую прибыль компании
- Автоматизируйте часто повторяющиеся операции

NOT SO GOOD

<https://aws.amazon.com/message/41926/>

Работы по автоматизации возможно полностью завершить в ограниченные сроки

- Реализации автоматизаций часто имеют недостатки
- Прогресс технологий: регулярно появляются новые подходы
- Постоянное развитие (держат команду в тонусе)
- Постоянные улучшения (правильный маркетинг)

FALSE

Маркетинговая стратегия

Опросники для клиентов:

- Насколько вы удовлетворены нашим сервисом?
- Какие улучшения вы бы хотели?
- Знаете ли вы о новых сервисах или улучшениях в существующих?
Попробуете?



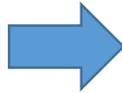
=> Вместо опросников спрашивайте об улучшениях, предлагайте новые фичи

Планы на будущее и хотелки клиентов

- Kubernetes
- Развитие сервиса оповещений
- Предиктивная аналитика возможных проблем
- Интерактивные письма
- Чатботы
- ...



План (мечта): создать команду автоматизации



Автоматизация не потребует много ресурсов на её поддержку

False

Подход с автоматизацией сэкономит много денег

Depends

Автоматизация всегда предпочтительнее ручной работы

Depends

Работы по автоматизации возможно полностью завершить в некие ограниченные сроки

False

Что мы получили в результате
применения автоматизации?

Парадокс автоматизации

До	После
Ручные работы на проде	Разработка автоматизации
Усилия по поддержке прода	Усилия по поддержке автоматизации
Цена ошибки в проде	Цена ошибки автоматизации
Ручное тестирование в песочнице	Разработка и поддержка тестов
Количество людей на поддержке	Количество людей на поддержке/разработке не уменьшается



Особенности и преимущества автоматизации

Автоматизация позволяет меньше жарить:

- Проблемы переезжают от клиента в автоматизацию
- Проблемы автоматизации решаются в спокойном режиме
- Клиенты вовремя получают оповещения

Больше автоматизации – больше усилий по её разработке/поддержке.

Автоматизация помогает экономить при правильном применении.

END OF STRESS ZONE

«Ради кофе можно пойти на всё. Даже на работу» – Билл Гейтс

Спасибо!



Антон Рымкус

 [@rymkus](https://twitter.com/rymkus)



Оксана Некрасова

 [@Oks_Nekrasova](https://twitter.com/Oks_Nekrasova)